DEVELOPMENT OF A TWO-LINK FLEXIBLE MANIPULATOR

MOHD ZAIDI BIN MOHD TUMARI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Mechatronics and Automatic Control)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

NOVEMBER 2009

Specially dedicated to
My beloved mother and father

ACKNOWLEDGEMENTS

Thanks to God because I have successfully completed my project. Since I'm doing my project, I was in contact with many people, researchers, academicians and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Associate Professor Dr. Zaharuddin Bin Mohamed, for encouragement, guidance, critics and friendship. Without his continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Pendidikan Sultan Idris (UPSI) for funding my Master study. Without UPSI financial sponsors in my Master study, certainly, I'm unable to continue my research project.

My sincere appreciation also extends to all my friends and others who have provided assistance at various occasions. Especially to PhD students, Mr Khairuddin and Mr Nasaruddin Ahmad for their views and tips. Further appreciations to my housemate, Nazrul Fariq Makmor and undergraduate students, Mazlan Muhammad and Noraide Yusop for their helps and cooperation. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members.

ABSTRACT

Flexible manipulator systems exhibit many advantages than the rigid robot systems. Flexible manipulator systems offered less overall costs, less material, small actuators, light in weight and consume less power than the traditional robot (rigid-arm). With the Two-Link Flexible Manipulator (TLFM) systems, various tasks can be done compare to one-link systems. However, this TLFM is rarely used in industry because of the difficulty in controlling the systems. Therefore, this thesis is to develop the mechanical structures of the TLFM and to design the real-time controller for this system so that it can operate smoothly and accurate. The mechanical structures of this two-link flexible manipulator consist of two DC motors, two flexible links, base and couplings. The hardware is interfaced with the computer through the data acquisition card and several analog to digital converter (ADC) and digital to analog converter (DAC). The Real-Time Window Target (RTWT) Toolbox in MATLAB/SIMULINK software is used for controlling the system. The feedback signal from encoder is used for real-time controller development. This flexible manipulator only considered in horizontal plane only.

ABSTRAK

Sistem manipulasi mudah lentur mempunyai banyak kelebihan berbanding sistem robot tegar**.** Antara kelebihannya ialah ia memerlukan kos keseluruhan yang rendah, kurangnya penggunaan material, penggerak yang kecil, bersaiz kecil dan penggunaan kuasa yang minima berbanding sistem tegar. Dengan sistem manipulasi mudah lentur dua lengan, pelbagai tugas boleh dilakukan berbanding sistem manipulasi satu lengan. Walaubagaimanapun, sistem dua lengan ini amat jarang digunakan di industri kerana kesukaran untuk mengawalnya. Maka, tesis ini adalah untuk menghasilkan struktur mekanikal sistem manipulasi mudah lentur dua lengan dan mereka sistem kawalan masa-benar supaya sistem ini dapat beroperasi dengan lancar dan tepat. Struktur mekanikal untuk sistem ini terdiri daripada dua DC motor, dua lengan mudah lentur, tapak dan pengikat. Perkakasan dihubungkan dengan komputer melalui kad capaian data dan beberapa penukar analog ke digital dan digital ke analog. Real-Time Window Target (RTWT) Toolbox di dalam perisian MATLAB/SIMULINK digunakan untuk mengawal sistem ini. Isyarat daripada pengesan kedudukan motor pula digunakan untuk membangunkan kawalan masa-benar. Sistem manipulasi mudah-lentur ini hanya beroperasi pada paksi mendatar sahaja.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	DECLARATION		ii
	DEDICATION		iii
	ACKNOWLEDGEMENTS		iv
	ABSTRACT		v
	ABSTRAK		vi
	TABLE OF CONTENTS		vii
	LIST OF TABLES		x
	LIST OF FIGURES		xi
	LIST OF ABBREVIATIONS		xiv
	LIST OF SYMBOLS		xv
	LIST OF APPENDICES		xvi
1	**INTRODUCTION**		
	1.1	Background of Project	1
	1.2	Problem Statement	3
	1.3	Objectives of the Project	3
	1.4	Scope of Project	3
2	**LITERATURE REVIEW**		
	2.1	Background of Flexible Manipulator	5
	2.2	Feedfoward Control	6
	2.3	Feedback Control	9
3	**METHODOLOGY**		
	3.1	TLFM System	13
	3.2	Data Acquisition System	15

	3.3	Real-Time System	16
	3.4	Control System	17
4	**ELECTRONICS SPECIFICATION**		
	4.1	Personal computer (PC)	19
	4.2	MATLAB/SIMULINK Software	19
	4.3	National Instrument (NI) PCI6221	20
	4.4	Measurement Computing (MS) PCI QUAD04	21
	4.5	National Instruments (NI) SCC 2345	22
	4.6	Signal Conditioning Analog Output Card	22
	4.7	SCB 37 Connection Box	23
	4.8	Maxon Motor Driver	23
	4.9	Maxon DC Motor	24
	4.10	Power supply	25
	4.11	Encoder	26
5	**EXPERIMENTAL SETUP**		
	5.1	Hardware Setup	27
		5.1.1 Mechanical Structure	27
		5.1.2 Hardware Connection	29
		5.1.3 Encoder Installation and Calibration	31
	5.2	Software Setup	32
		5.2.1 Real-Time Window Target Setup	32
		5.2.1.1 Installation and Configuration of RTWT	32
		5.2.1.2 Procedures of Creating Real-Time Applications	35
		5.2.2 Simulink Block Diagram for TLFM	47
	5.3	PID Tuning	49
6	**RESULT AND DISCUSSION**		
	6.1	Open Loop (Feedforward) System	51
	6.2	The Closed Loop System without PID Controller	53
	6.3	The Closed Loop System with PID Controller	56
	6.4	Discussion	59

7 CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion 61

7.2 Recommendations 61

REFERENCES 63

Appendices A-J 65-85

LIST OF TABLES

TABLE NO.	TITLE	PAGE
4.1	The specification of the DC-motor	24
5.1	The initial PID controller parameters for DC-motors	50
6.1	Experimental Result	60

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	The two-link flexible manipulator (TLFM)	2
3.1	The research methodology	13
3.2	The description of a TLFM system	14
3.3	Block diagram of the TLFM system	18
4.1	The PCI 6221	20
4.2	The PCI QUAD04	21
4.3	The SCC 2345	22
4.4	The signal conditioning analog output card	23
4.5	The SCB 37 connection box	23
4.6	The Maxon motor driver	24
4.7	The first motor	25
4.8	The second motor	25
4.9	The power supply	25
4.10	The HEDL-5540 encoder	26
4.11	The HEDS-5540 encoder	26
5.1	The complete mechanical structure of TLFM	28
5.2	The actual image of TLFM	28
5.3	The hardware connection	29
5.4	The actual image of hardware setup	30
5.5	Encoder installation on the motor	31
5.6	Simulink model rtvdp.mdl window	33
5.7	Scope windows	34
5.8	Created a new Simulink model	35
5.9	An empty Simulink window	35
5.10	Block parameters dialog block	36
5.11	Block parameters of signal generator	37

5.12	I/O Block parameters dialog box	38
5.13	Display message dialog box	38
5.14	Scope parameters dialog box	39
5.15	Scope parameters	40
5.16	Scope window for y-axis dialog box	40
5.17	Scope properties axis 1 dialog box	41
5.18	Completed Simulink block diagram	41
5.19	Solver pane dialog box	42
5.20	Solver pane dialog box for hardware implementation	43
5.21	System target file browser	44
5.22	Real-time workshop pane	44
5.23	Connect to target from the simulation menu	46
5.24	Start real-time code from simulation menu	46
5.25	The Simulink block diagram for TLFM	47
5.26	Stable oscillation	49
6.1	The angular displacement of first motor with positive input (open loop)	52
6.2	The angular displacement of first motor with negative input (open loop)	52
6.3	The angular displacement of second motor with positive input (open loop)	52
6.4	The angular displacement of second motor with negative input (open loop)	53
6.5	The angular displacement of first motor with positive rotation (closed loop)	54
6.6	The angular displacement of first motor with negative rotation (closed loop)	54
6.7	The angular displacement of second motor with positive rotation (closed loop)	55
6.8	The angular displacement of second motor with negative rotation (closed loop)	55
6.9	The angular displacement of first motor with positive input (closed loop with PID)	56

6.10 The angular displacement of first motor with negative input (closed loop with PID) 57
6.11 The angular displacement of second motor with positive input (closed loop with PID) 58
6.12 The angular displacement of second motor with negative input (closed loop with PID) 58

LIST OF ABBREVIATIONS

RLM	-	rigid link manipulators
TLFM	-	two-link flexible manipulator
ADC	-	analog to digital converter
DAC	-	digital to analog converter
RTWT	-	Real-time window target
PID	-	proportional integral differential
PD	-	proportional derivative
LQG	-	linear quadratic Gaussian
LQR	-	linear quadratic regulator
MRLC	-	multirate repetitive learning control
MIMO	-	multi input multi output
PSD	-	position sensitive detector
DOF	-	degree of freedom
DAQ	-	data acquisition
DAS	-	data acquisition system
PC	-	personal computer
PCI	-	peripheral component interconnect
TTL	-	transistor-transistor logic
RAM	-	random access memory
RTC	-	real-time computing
I/O	-	input output
SSE	-	steady state error
DC	-	direct current

LIST OF SYMBOLS

e	-	tracking error
r	-	desired output
u	-	input
y	-	measured output
Kp	-	proportional gain
Ki	-	integral gain
Kd	-	differential gain
Ts	-	time settling
Td	-	time delay
Tc	-	time constant
l	-	length
τ	-	torque
E	-	modulus young
I	-	moment of inertia
	-	mass density per unit volume
A	-	cross section
□	-	angle 1
	-	angle 2
XY	-	stationary axis
X_1Y_1	-	moving coordinates 1
X_2Y_2	-	moving coordinates 2

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Measurement Computing PCI QUAD04	65
B	Signal Conditioning Analog Output Card	68
C	Maxon Motor Driver	70
D	Maxon DC-Motor RE40-40mm	72
E	Maxon DC-Motor RE35-35mm	74
F	Power Supply	76
G	Encoder HEDL-5540	79
H	Encoder HEDS-5540	81
I	Gantt Chart for semester 1	83
J	Gantt Chart for semester 2	85

CHAPTER 1

INTRODUCTION

1.1 Background of Project

Most of industrial sector use a robot applications as a tool to help them in production. The robot applications, known as an industrial robot, are helpful in doing various tasks such as displace the product, assembled parts, soldering and spray painting. Accurate operation, smooth movement and without vibration are the behaviors that demand in robot applications. The accurate positioning of the end-effectors is important to perform the certain tasks and to avoid a failure in the operation.

Traditionally, rigid link manipulators (RLM) are designed and built in a manner that maximizes stiffness to minimize vibration and allow for good positional accuracy with relatively simple controllers. High stiffness is achieved by using heavy material that limits the rapid motion of the manipulators, increase the sizes of actuators and boost energy consumption. Therefore, much cost is required to perform the traditional robot in industry. RLM are heavy and massive, and their load-carrying capacity is limited to avoid the vibration of the links. Because of that, the rigid robot is inefficient and slow when dealing in industry manufacture.

The Two-Link Flexible Manipulator (TLFM) system is introduced to overcome the problem of the RLM. This flexible manipulator made from the lightweight material and not rigid. Therefore, in general, the flexible manipulator exhibits more advantages than the rigid robots; required less material, lighter in

weight, consumes less power, required small actuators, more transportable, less overall cost and higher payloads to robot weight ratio. It also safer to operate because of reduced inertia. Figure 1.1 shows the example of the TLFM.

Figure 1.1: The two-link flexible manipulator (TLFM)

Over the past two decades the need for high speed manipulation and high payload to weight ratio in robot manipulators has triggered a significant growth in the research and development of the flexible manipulators. Flexible manipulators are a suitable choice to realize such needs since they are light in weight require small sized actuators and low energy consumption. However, the use of these flexible manipulators at a high speed poses challenging problems in designing their control system. The control system must be designed not only for precise tip positioning but also for suppressing the vibrations arising due to the flexible nature of the manipulator.

1.2 Problem Statement

Flexible manipulators are less expensive to manufacture and safer to operate because of reduced inertia. It also can operate in high speed movement with the same or even better precision compared to RLM. However, the manipulators have a vibration problem, become more flexible and more difficult to control accurately due to light in weight and low stiffness. The control difficulty is caused by fact that since the manipulators are distributed, a large number of flexible modes are required to accurately model its behaviors. Further complications arise because of the highly nonlinear nature of the system. Therefore, flexible manipulators have not been favored in industries, as the manipulator is required to have reasonable end-point accuracy in response to input commands. In this respect, accurate models and efficient controllers have to be developed.

1.3 Objectives of the Project

The objectives of this project are:

i. To develop a prototype of a two-link flexible manipulator.
ii. To implement encoder as a feedback signal for controlling the system.
iii. To design and implement a real-time controller for two- link flexible manipulator.

1.4 Scope of the Project

This project is focuses on the design of the mechanical structure of a TLFM and design of real-time controller of the system. The controller is design based on MATLAB/SIMULINK software. Therefore, the interface between hardware and software also required to operate the system. The hardware is interfaced with the computer through the data acquisition card and several analog to digital converter (ADC) and digital to analog converter (DAC). The MATLAB/SIMULINK software is

used to simulate the system where the Real Time Window Target (RTWT) as a main toolbox to implement the system. PID controller has been used as a controller and the Ziegler-Nichols method is used for tuning. The scope of the project also includes the research on encoder as a feedback signal for design of controller. The two degrees of freedom flexible manipulator only operates in horizontal plane only.

CHAPTER 2

LITERATURE REVIEW

2.1 Background of Flexible Manipulator

According to M.A Ahmad (2006), robotic manipulators are design to maximize, in an attempt to minimize system vibration and achieved good positional accuracy. In order to make the manipulator become more stable and accurate, the heavier design of robotic manipulator has been done. Unfortunately, this will effect to the speed of robot operation. Moreover, these design hard to provide high ratio of the load to the arm weight considerably and also increase the energy consumption. In order to overcome these problems, researchers begin to design the flexible manipulator system.

Many researches on designing the flexible manipulator, there are two crucial problems that must be resolve. First is to make the flexible manipulator system to achieve and maintain accurate positioning. Another issue is the system must also stabilize the vibrations that are naturally excited. The dynamic formulation of flexible manipulators leads a set of complicated partial differential equations and cannot be solved analytically. In research by Mohamed and Tokhi (2002), highlight that the control mechanism that account for both the rigid body and flexural motion of the system are required for the respect of such complicated equation.

Gulay Oke and Yorgo Istefarwpulos (2001) have mention that the various approach of technique has been proposed for the control of flexible manipulators in the literature. Some examples are PD control, end-point acceleration feedback, state

feedback, model reference adaptive control, feedback linearization, input frequency pre-shaping methods, optimal control techniques like LQG compensator, minimum-time control, bang-bang control, singular perturbation techniques, robust control techniques, like H∞ controllers, sliding mode controllers, intelligent control methods like fuzzy logic control and neural networks. Z. Mohamed (2003) says that the vibrations control for flexible manipulator systems can be classified as feedforward and feedback control schemes .The fundamental problem with systems that vibrate is those motion transients excite the vibration.

2.2 Feedfoward Control

Feedforward control is relating an element within a control system which passes a controlling signal from a source in the control system's external environment, often a command signal from an external operator, to a load elsewhere in its external environment. A control system which has only feed-forward behavior responds to its control signal in a pre-defined way without responding to how the load reacts. This is a control technique that can be measured but not controlled. The disturbance is measured and fed forward to an earlier part of the control loop so that corrective action can be initiated in advance of the disturbance having an adverse effect on the system response. Feedforward control is always used along with feedback control because a feedback control system is required to track set point changes and to suppress unmeasured disturbances that are always present in any real process.

Works by M.A Ahmad (2006) have shown that the feedfoward control can be used for vibration control by measuring one or more frequencies that are excited by the motion transient. Due on this fact, the system without vibrations can be develop by using a modified command signal as an input. Furthermore, this method does not require any additional sensor or actuators and does not account for changes in the system once the input is developed. As a result of Mohamed and Tokhi (2002), the feedfoward control scheme is considered base on input shaping and low pass filtering techniques.

Ke-Ping Liu (2007) has proposed an input shaping technique in order to reduce the vibration of flexible manipulator. In this approached, the input shaping is implemented by convolving the reference command with a sequence of impulses. The design objective is to choose the impulse amplitudes and time locations so that the shaped (filtered) command reduces the detrimental effects of system flexibility. If the impulse amplitudes and time location are chosen correctly, then any function can be used as the original command and the subsequent shaped command will result in zero vibration.

Other previous research by Mohd Asyraf Ahmad and Zaharuddin Mohamed (2007) have developed a hybrid control schemes with the applications of input shapers in command shaping techniques for vibration control and input tracking of a flexible robot manipulator. A constrained planar single-link flexible manipulator is considered and the dynamic model of the system is derived using the assume mode method. To study the effectiveness of the controllers, initially a Linear Quadratic Regulator (LQR) is developed for control of rigid body motion. This is then extended to incorporate input shaper control schemes for vibration control of the system. The input shapers are designed based on the properties of the system. Simulation results of the response of the manipulator to the shaped inputs are presented in time and frequency domains. Performances of the hybrid control schemes are examined in terms of level of input tracking capability, vibration reduction, time response specifications and robustness to parameters uncertainty.

On study by W.S. Jang, K.S. Kim and S.K. Lee (2003), they have used a three-step input method, whereby a flexible robot arm is moved from one position to another with a minimum of residual vibration when the arm reaches its defined endpoint. This work is concerned with defining a simple practical method to utilize step inputs to achieve optimum response. The optimum response is achieved by using a self adjusting input command function that is obtained during real-time processing. These methods are computationally intensive and hence are not cost effective. These techniques can be divided into two broad categories, dependent upon one's viewpoint. One category is based on closed-loop feedback control techniques, which increases damping by a limited amount. If the inherent damping is very low, this increase may

be insufficient to adequately improve the response. The other category is based on input command shaping techniques, which has examined the transient vibration of robot manipulators in terms of frequency content of the system inputs and outputs. This approach inherently assumes that the system inputs are not actually transient, but are one cycle of a repeating wave form.

In addition, Sungsoo Rhim and Ai-Ping Hu (2003) using a tip controller based on multirate repetitive learning control (MRLC), in command shaping filter is used to reshape the reference trajectory. For a typical trajectory, the MRLC based tip controller has been shown to be capable of significantly reducing tip tracking error though exhibiting relatively residual tip vibrations and requiring an accompanying large actuation effort. A tip reference trajectory reshaped by a properly-designed command shaping filter smoothes the trajectory's acceleration profile and avoids excitation of the system resonance. Numerical simulations demonstrate that the result is a drastic reduction in residual tip vibrations and the required actuation effort of the MRLC-based tip controller.

Other control schemes that successfully implement the feedfoward technique are input-output linearization using fuzzy logic control by Y.Sun, R.Wu and F. Cheng (2007). It based on input-output linearization strategy, a model of the flexible manipulator can be decomposed into an input-output subsystem and a zero-dynamics subsystem. A fuzzy control is used to input-output subsystem. By analyzing the relationship between the Eigen values of the zero-dynamics subsystem and the designed parameters in the output of the redefined system, the designed parameters are determined, and guarantee the zero-dynamics to be asymptotically stable at equilibrium point, thus guarantee the whole flexible manipulator system to be asymptotically stable.

2.3 Feedback Control

A closed-loop controller uses feedback to control states or outputs of a dynamical system. Thus, it has ability to overcome the problem of the open-loop system. Its name comes from the information path in the system: process inputs have an effect on the process outputs, which is measured with sensors and processed by the controller; the result is used as input to the process, closing the loop.

The PID controller is probably the most-used feedback control design, being the simplest one. PID means Proportional-Integral-Derivative, referring to the three terms operating on the error signal to produce a control signal. If u(t) is the control signal sent to the system, y(t) is the measured output and r(t) is the desired output, and tracking error.

((− ((2.1)

PID controller has the general form that is:

$u($ (d — (2.2)

The desired closed loop dynamics is obtained by adjusting the three parameters Kp, Ki and Kd, often iteratively by “tuning” and without specific knowledge of a plant model. Stability can often be ensured using only the proportional term. The integral term permits the rejection of a step disturbance. The derivative term is used to provide damping or shaping of the response. PID controllers are the most well established class of control systems: however, they cannot be used in several more complicated cases, especially if MIMO systems are considered.

A. Mahmood, B.Bhikkaji and S. O. R. Moheimani (2007) illustrate that collocated sensors such as a shaft encoder or a strain gauge placed at the root end of the flexible manipulator have also been used in the control of flexible manipulators. In his work an experimental flexible manipulator setup was fabricated to represent a typical flexible manipulator. A control strategy using both collocated and no

collocated measurement was developed for vibration suppression as well as for the position control of the flexible manipulator. A shaft encoder and a position sensitive detector (PSD) were used for obtaining collocated and no collocated measurements respectively. The use of each measurement would provide a specific benefit. The control strategy consists of two feedback loops. In the first loop, a polynomial based pole placement controller, was designed to add damping to the flexible manipulator and in the second loop, an integral controller was designed for precise tip positioning. The input to the polynomial-based controller is the collocated hub angle measurement, and the input to the integral controller is the tip position.

Previous study using the strain gauges as a feedback sensor has been carried out by Jan-Peter Hauschild (2003) in controlling flexible link robotic manipulator in zero gravity. Strain gauges can be used to achieve the requirement of the dynamic measurement of the deflection and angle of the endpoint. Since a flexible link performs not only at its fundamental mode but also at its higher modes, more than one measurement of strain must be done along the link to provide accurate dynamic result.

Other study done by Ram Somaraju (2003) presents an approximate small-deflection model for the flexible beam using a truncated Eigen function expansion. The deflection and angular position of the beam were measured using strain gauges and an optical encoder respectively. This model was experimentally verified by measuring the frequency response of the beam. This paper in on the experimental setup and model developed proportional-derivative (PD) and linear quadratic regulator controllers were developed for beams that carry no payload.

Besides that, other controller scheme for controlling double flexible link is by using fuzzy tune PID as described by Ahmad Suryo Arifin *et al.* (2007). Fuzzy tune PID has advantage because the design like PID, but each value of Kp, Ki and Kd are determined by fuzzy. Fuzzy tune PID more suitable to nonlinear system and relatively more robust. The plant itself has nonlinear characteristic, so fuzzy tune PID seems to be the right choice for the plant. The system presentation consist of the 2-Degree of freedom (DOF) serial flexible link robot. This research uses strain gauge and optical encoder as a feedback signal.

In addition, the study by Noraide Md Yusop (2009) has developed a direct strain feedback with PID controller in controlling the flexible manipulator. Strain gauge is the main sensor used as a strain measurement for giving a feedback to the system. The strain measurement also has been used as displacement sensor at the endpoint of the link. The displacement was used to observe the performance of the system. The performances of the controllers are assessed in terms of the input tracking capability (desired position) and vibration reduction as compared the original system.

Meanwhile, the research by Mazlan Muhammad (2009) has implemented an accelerometer for measurement of end point acceleration of a single link of a flexible manipulator. The project involves the process of installation, choosing location for accelerometer on the beam and also measurement of acceleration. The implementation of the proportional-integral-derivative controllers (PID controllers) as a feedback controller to control of hub angle motion of flexible manipulator system is also studied. Ziegler-Nichols method is used to determine the best value for the PID controllers gain in order to operate the system properly and accurate.

CHAPTER 3

METHODOLOGY

Several methodologies are used in order to achieve the entire project objectives. First, the concept and idea of the TLFM have been researched from previous researches. It is important to be familiar with the flexible manipulator system in order to implement this project. After that, mechanical structure of two-link flexible manipulator has been researched to gain information about the material that must be used for this flexible links. Then, the design of mechanical structure is carried out by using the particular software and then it has been fabricated according to the design. Next, is studied on interfacing between hardware and software in order to control the system. The software that has been used in this project is MATLAB/SIMULINK that is responsible as a main controller for the system and also to design the real-time controller, so that the system can operate smoothly and accurately. Then, several experiments have been carried out in order to get the optimum performance for the system. The research methodology is summarized in Figure 3.1.

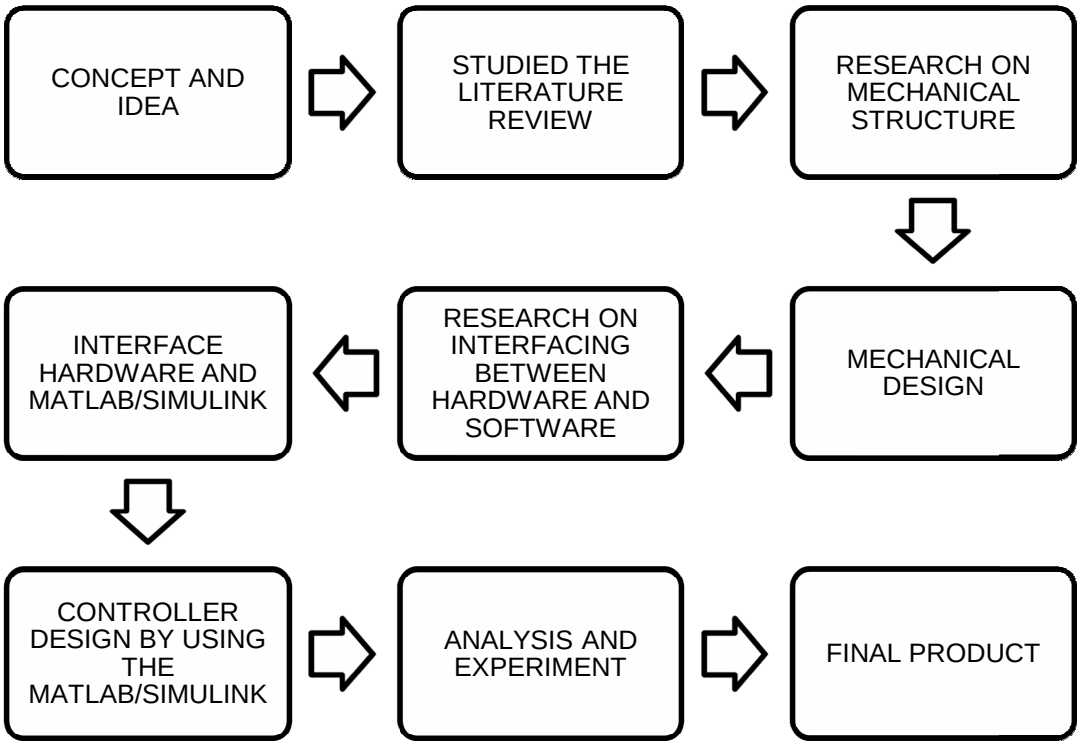
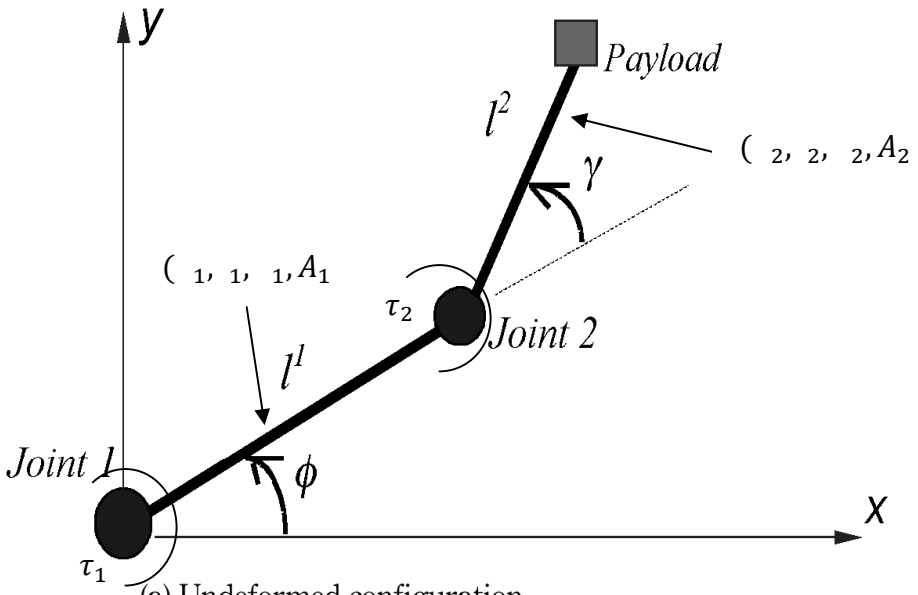


Figure 3.1: The research methodology

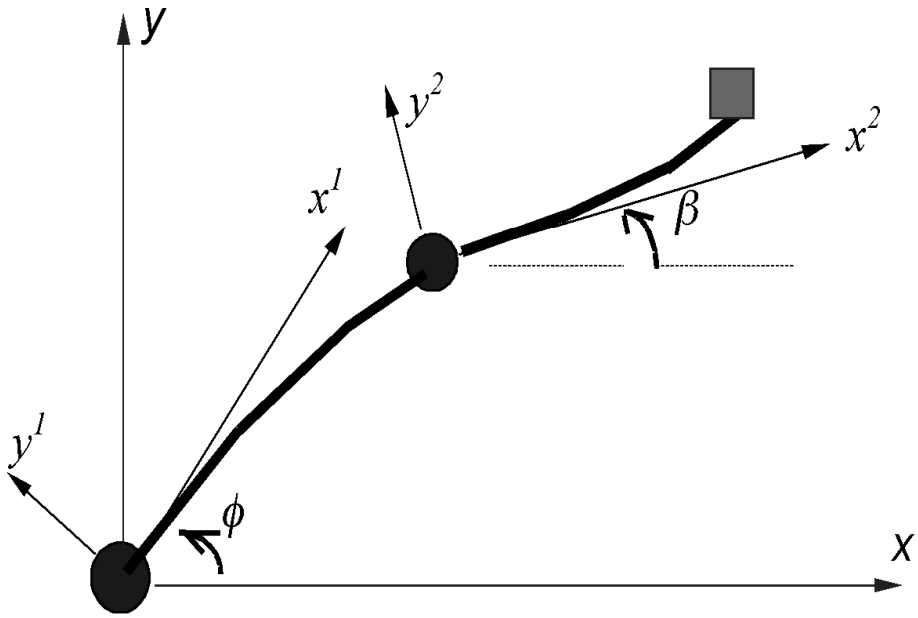
3.1 TLFM System

A description of the TLFM system is shown in Figure 3.2. XY axis represents a stationary axis while X_1Y_1 and X_2Y_2 axis represent the moving coordinates for link-1 and link-2 respectively. The length of link-1 and link-2 is represented by l_1 and l_2 respectively. τ_1 and τ_2 represent the applied torque for link-1 and link-2 respectively while ($_1$, $_1$, $_1$, A_1 and ($_2$, $_2$, $_2$, A_2 represent Young modulus, second moment of inertia, mass density per unit volume and a cross section of manipulator for link-1 and link-2 respectively. The motion of manipulator confine to the XY plane. The rotation of X_1Y_1 relative to XY plane is described by angle ϕ while rotation of X_2Y_2 relative to XY plane is described by angle . Due to beam is long and slender, transverse shear and rotary inertia effects are neglected. This allows the use of Bernoulli-Euler beam theory to model the elastic behavior of the manipulator. The manipulator is assumed to be stiff in vertical bending and torsion, thus allowing it to vibrate dominantly in the horizontal direction. The beam is considered to have a constant cross section and

uniform material properties throughout. Figure 3.2 (a) and (b) show the un-deformed and deformed configuration respectively.



(a) Undeformed configuration.



(b) Deformed configuration.

Figure 3.2: The description of a TLFM system (a) un-deformed configuration (b) deformed configuration

3.2 Data acquisition system

Data acquisition system is responsible for interfacing between hardware and software. Data acquisition is the sampling of the real world to generate data that can be manipulated by a computer. Sometimes abbreviated DAQ or DAS, data acquisition typically involves acquisition of signals and waveforms and processing the signals to obtain desired information. The components of data acquisition systems include appropriate sensors that convert any measurement parameter to an electrical signal, which is acquired by data acquisition hardware.

Acquired data are displayed, analyzed, and stored on a computer, either using vendor supplied software, or custom displays and control can be developed using various general purpose programming languages such as BASIC, C, Fortran, Java, Lisp, Pascal. Specialized programming languages used for data acquisition include, EPICS used to build large scale data acquisition systems, LabVIEW, which offers a graphical programming environment optimized for data acquisition and MATLAB provides a programming language but also built-in graphical tools and libraries for data acquisition and analysis.

Data acquisition begins with the physical phenomenon or physical property of an object (under investigation) to be measured. This physical property or phenomenon could be the temperature or temperature change of a room, the intensity or intensity change of a light source, the pressure inside a chamber, the force applied to an object, or many other things. An effective data acquisition system can measure all of these different properties or phenomena.

A transducer is a device that converts a physical property or phenomenon into a corresponding measurable electrical signal, such as voltage, current, change in resistance or capacitor values, etc. The ability of a data acquisition system to measure different phenomena depends on the transducers to convert the physical phenomena into signals measurable by the data acquisition hardware. Transducers are synonymous with sensors in DAQ systems. There are specific transducers for many different applications, such as measuring temperature, pressure, or fluid flow. DAQ

also deploy various signal conditioning techniques to adequately modify various different electrical signals into voltage that can then be digitized using ADCs.

DAQ hardware is what usually interfaces between the signal and a PC. It could be in the form of modules that can be connected to the computer's ports (parallel, serial, USB, etc...) or cards connected to slots (PCI, ISA) in the mother board.. The cable between this Box and the PC is expensive due to the many wires and the required shielding and because it is exotic. DAQ-cards often contain multiple components (multiplexer, ADC, DAC, TTL-IO, high speed timers, RAM). These are accessible via a bus by a micro controller, which can run small programs.

3.3 Real-Time System

Real-time computing (RTC) is the study of hardware and software systems that are subject to a "real-time constraint"—i.e., operational deadlines from event to system response. By contrast, a non-real-time system is one for which there is no deadline, even if fast response or high performance is desired or even preferred. The needs of real-time software are often addressed in the context of real-time operating systems, and synchronous programming languages, which provide frameworks on which to build real time application software.

The term real-time derives from its use in early simulation. While current usage implies that a computation that is 'fast enough' is real-time, originally it referred to a simulation that proceeded at a rate that matched that of the real process it was simulating. Analog computers, especially, were often capable of simulating much faster than real-time, a situation that could be just as dangerous as a slow simulation if it were not also recognized and accounted for.

A system is said to be real-time if the total correctness of an operation depends not only upon its logical correctness, but also upon the time in which it is performed. The classical conception is that in a hard or immediate real-time system, the

completion of an operation after its deadline is considered useless - ultimately, this may lead to a critical failure of the complete system. A soft real-time system on the other hand will tolerate such lateness, and may respond with decreased service quality.

Hard real-time systems are used when it is imperative that an event is reacted to within a strict deadline. Such strong guarantees are required of systems for which not reacting in a certain window of time would cause great loss in some manner, especially physically damaging the surroundings or threatening human lives (although the strict definition is simply that missing the deadline constitutes failure of the system).

Soft real-time systems are typically used where there is some issue of concurrent access and the need to keep a number of connected systems up to date with changing situations; for example software that maintains and updates the flight plans for commercial airliners. The flight plans must be kept reasonably current but can operate to a latency of seconds. Live audio-video systems are also usually soft real-time; violation of constraints results in degraded quality, but the system can continue to operate.

3.4 Control System

The setup of the control system is schematically shown in Figure 3.3. It consists of two main blocks (PC and Flexible Manipulator) that are connected together. The control algorithms are programmed in Matlab/Simulink, compiled with the RTWT toolbox. RTWT are includes an analog input and output that provides connection between the physical I/O board (PCI 6221 & PCI QUAD 04) and the real-time model. For example, the rotation of the DC motor can be controlled by supply certain voltage and frequency from the signal generator to the analog output in the Simulink. The response from the analog input also can be displayed in the scope

provided in Simulink. From these two ways of communication, a closed loop system with PID and encoder feedback can be built to control the rotation of the TLFM.

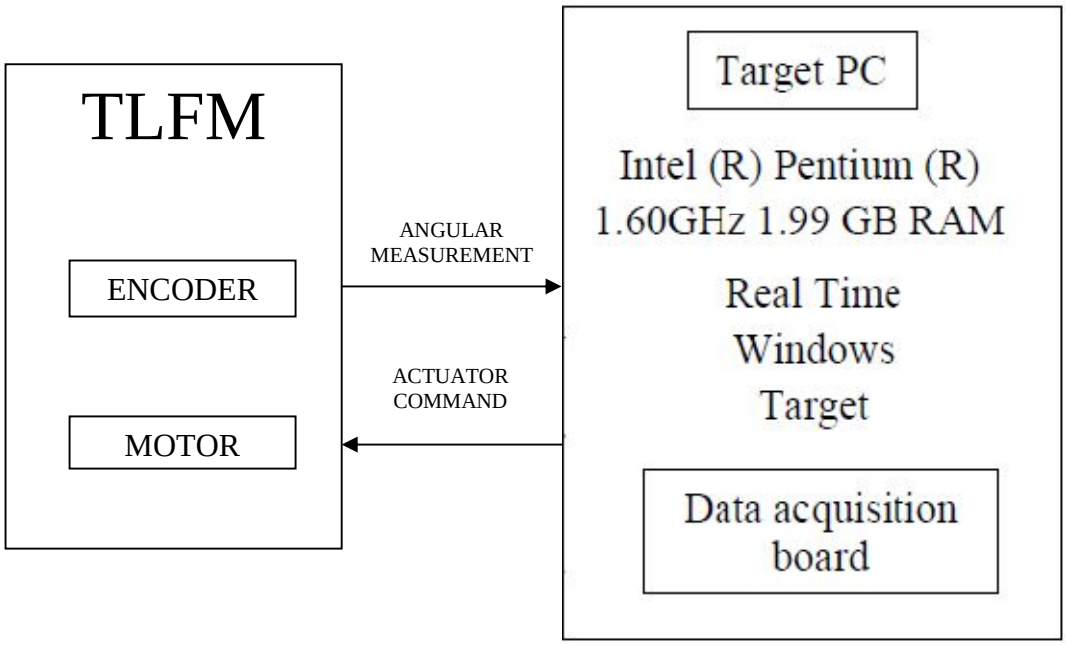


Figure 3.3: Block diagram of the TLFM system

CHAPTER 4

ELECTRONICS SPECIFICATION

This project involves the hardware and software. Thus, the interfacing between hardware and software is needed. This interfacing used a number of electronic components. The software that has been used in this research is MATLAB/SIMULINK while for the hardware consists of personal computer (PC), National Instrument (NI) PCI 6221, Measurement Computing (MC) PCI QUAD04, NI SCC 2345, SCB 37 connection box and encoder. All the datasheets for hardware can be viewed in Appendix.

4.1 Personal Computer (PC)

PC is used as a main controller of the systems. It responsible to sends and receives data from the hardware. The specifications of the computer are given below:

- Intel (R) Pentium (R) Dual CPU E2140 @ 1.60GHz
- 1.99 GB of RAM

4.2 MATLAB/SIMULINK Software

MATLAB/SIMULINK is used as a controller for systems. The signal from the hardware will simulate by MATLAB as a parameters for the controller. The Real

Time Window Target Toolbox is a main tool to implement the systems. The version of the MATLAB is:

- Version 7.6.0.324 (R2008a)
- Released in February 10, 2008

4.3 National Instrument (NI) PCI6221

PCI 6221 is a multifunctional Data Acquisition Card (DAQ) for synchronous an input and output signal from the hardware to computer. Figure 4.1 shows the actual image of PCI6221. The specifications of the DAQ card are listed below:

- Two 16-bit analog outputs (833 kS/s); 24 digital I/O; 32-bit counters
- NIST-traceable calibration certificate and more than 70 signal conditioning options
- Correlated DIO (8 clocked lines, 1 MHz)
- NI-MCal calibration technology for increased measurement accuracy
- Select high-speed M Series for 5X faster sampling rates or high-accuracy M Series for 4X resolution.
- NI-DAQmx driver software and NI LabVIEW Signal Express interactive data-logging software.

Figure 4.1: The PCI 6221

4.4 Measurement Computing (MS) PCI QUAD04

This card is used as interfaces between the MATLAB and encoders in the motor. It reads the counts from the encoder and MATLAB is responsible to convert it into position of the motor. The actual image of this card is shown in Figure 4.2. It has 4 channels and the specs of this card are shown below:

- Extremely low cost
- Interfaces to industry standard encoders
- 4 input channels
- Inputs are configurable for single-ended or differential mode
- An interrupt controller yields powerful interrupt management capabilities
- Quad receivers can tolerate long lines in noisy environments
- Uses integrated digital filtering and 8-bit counter pre-scale
- Programmable index functionality provided
- Programmable multiple-counter modes possible including non-quadrature cascaded counting up to 96 bit
- Each 24-bit counter can count in either binary or BCD modes

Figure 4.2: The PCI QUAD04

4.5 National Instruments (NI) SCC 2345

SCC 2345 as shown in Figure 4.3, is a portable low cost signal conditioning. It functions as analog/digital output and analog input. It consists of several channels for signal conditioning card. Below are the details:

- SCC-2345 connector block
- SCC-2345 with configurable connectors
- M Series and E Series DAQ device direct cabling
- Portable, low-profile packaging

Figure 4.3: The SCC 2345

4.6 Signal Conditioning Analog Output Card

The analog output is responsible to convert the digital output from computer to the analog output for motor driver. Figure 4.4 shows the analog output card that placed on the SCC 2345. Below are the specs of this card:

- 1-channel modules
- 300 V working safety isolation (Category II)
- 0 to 20 mA output (SCC-CO20)
- ±10 V output (SCC-AO10)

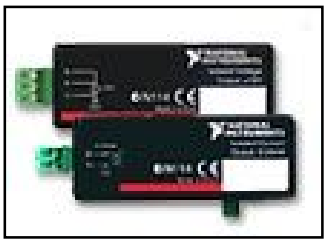


Figure 4.4: The signal conditioning analog output card

4.7 SCB 37 Connection Box

It functions as a connection box for PCI QUAD04 for easily connects the cable from the encoder to the PCI QUAD04. Figure 4.5 shows the image of SCB 37. The details are given below:

- Signal connection box
- 37-conductor
- Shielded

Figure 4.5: The SCB 37 connection box

4.8 Maxon Motor Driver

The 4-Q-DC servo-amplifier ADS in module housing is responsible to control the current to the DC motor in order to control the speed and the torque of the motor.

The datasheet of this motor driver is shown in appendix. Figure 4.6 shows the Maxon Motor Driver.

Figure 4.6: The Maxon motor driver

4.9 Maxon DC Motor

DC motor has been used as an actuator to the flexible arm. There are two DC motors that have been used in this project. Table 4.1 lists the specification of the both motor. The images of the first motor and second motor are shown in Figure 4.7 and Figure 4.8 respectively.

Table 4.1: The specification of the DC-motor

	Unit	Motor 1	Motor 2
Assigned power rating	W	150	90
Nominal voltage	V	24	24
No load speed	min-1	7580	7670
Stall torque	mNm	2280	1160
Max. continuous torque	mNm	170	93.3
Speed / torque gradient	rpm/mNm	3.33	6.79
No load current	mA	137	168
Starting current	A	75.7	39.7
Max. efficiency	%	90.9	84

Figure 4.7: The First Motor

Figure 4.8: The Second Motor

4.10 Power Supply

Power supply is used to supply the voltage/current to the motor driver and DC motor. The image of power supply is shown in Figure 4.9. The specifications of power supply that has been used are:

- PSH-6018A
- 1080W, Single Output, Programmable Switching D.C. Power Supply
- Wide Input Voltage Range and High Power Factor (P.F)
- High Efficiency and High Power Density
- Constant Voltage and Constant Current Operation
- Over Voltage, Over Current and Over Temperature Protection
- Low Ripple and Noise

Figure 4.9: The Power Supply

4.11 Encoder

Two encoders are used to measure the position of the motor which is HEDL-5540 and HEDS-5540. Both of them count about 500 counts per turn. Figure 4.10 shows the HEDL-5540 type for first motor while Figure 4.11 shows the HEDS-5540 type for second motor. Both datasheets can be view in appendix.

Figure 4.10: The HEDL-5540 encoder

Figure 4.11: The HEDS-5540 encoder

CHAPTER 5

EXPERIMENTAL SETUP

Several setups have been done to achieve the goals of this research project. The experimental setup can be divided into three parts which is hardware setup, software setup and PID Tuning. Those setups are to ensure the TLFM can be control properly.

5.1 Hardware Setup

5.1.1 Mechanical Structure

The mechanical structure has been fabricated by using the aluminum. The fabrication of mechanical structure has to be tidy enough and robust to avoid the problem in TLFM operation later. The complete mechanical structure of TLFM is shown in Figure 5.1 and the actual image of TLFM in Figure 5.2. The length and width of base is 300mm and 300mm respectively while the height is 230mm. The first link and a second link have a length about 500mm. The total weight of TLFM is 7.0 kilogram.

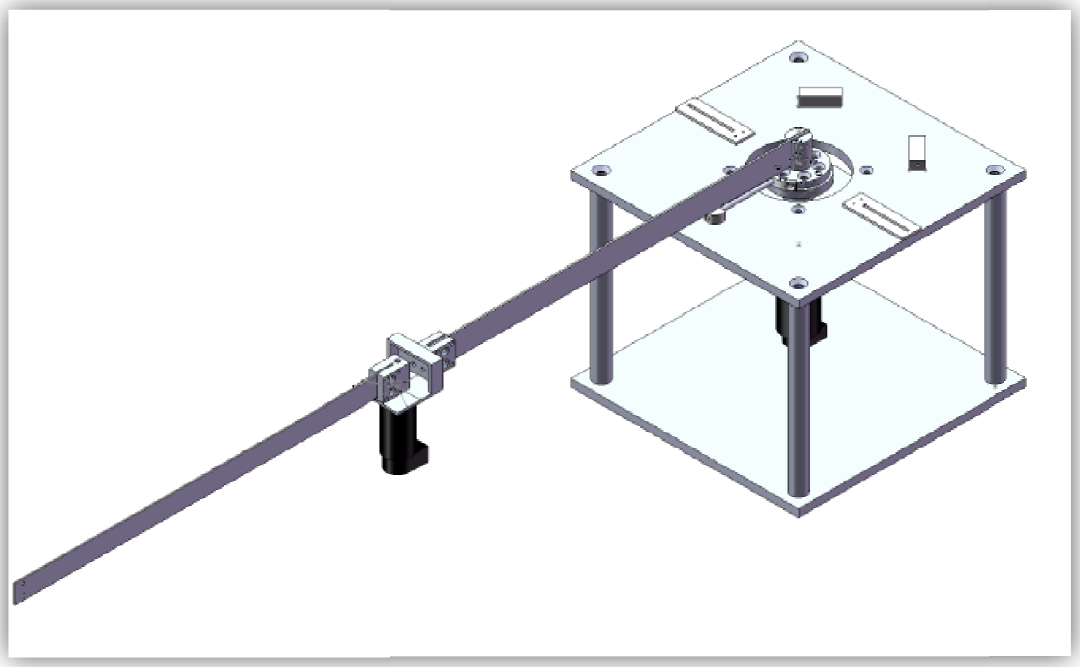


Figure 5.1: The complete mechanical structure of TLFM

Figure 5.2: The actual image of TLFM

5.1.2 Hardware Connection

The hardware setup is for interfacing between hardware and software. It involves the connection between TLFM, electronic components and the computer. Figure 5.3 illustrates the hardware connection while Figure 5.4 shows the actual hardware setup.

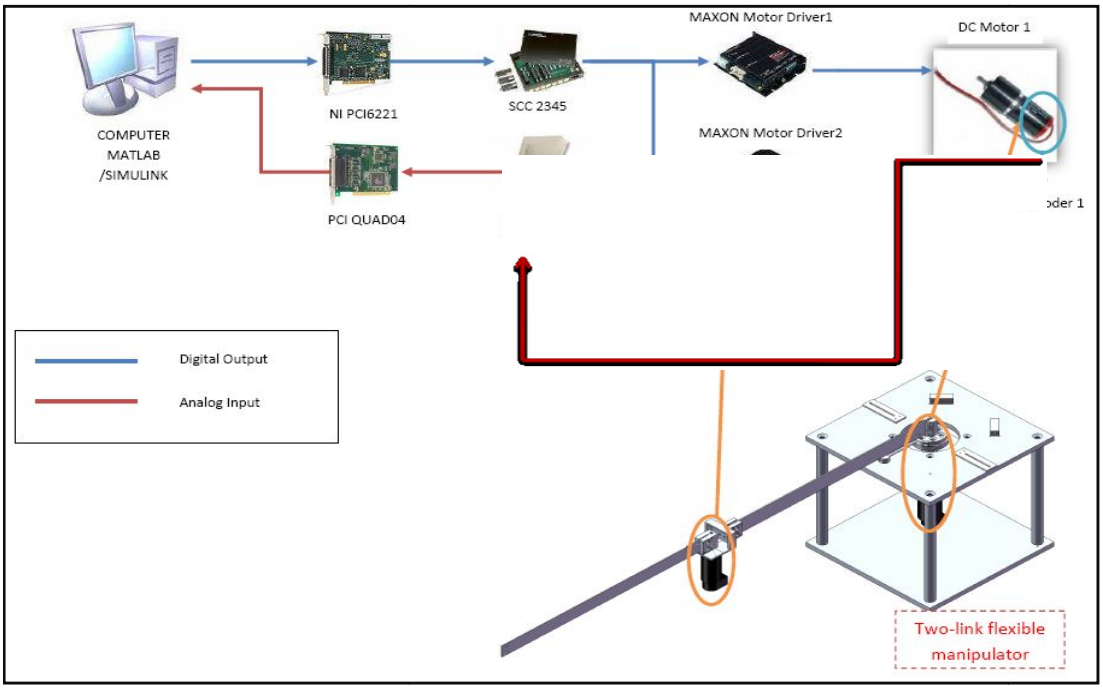


Figure 5.3: The hardware connection

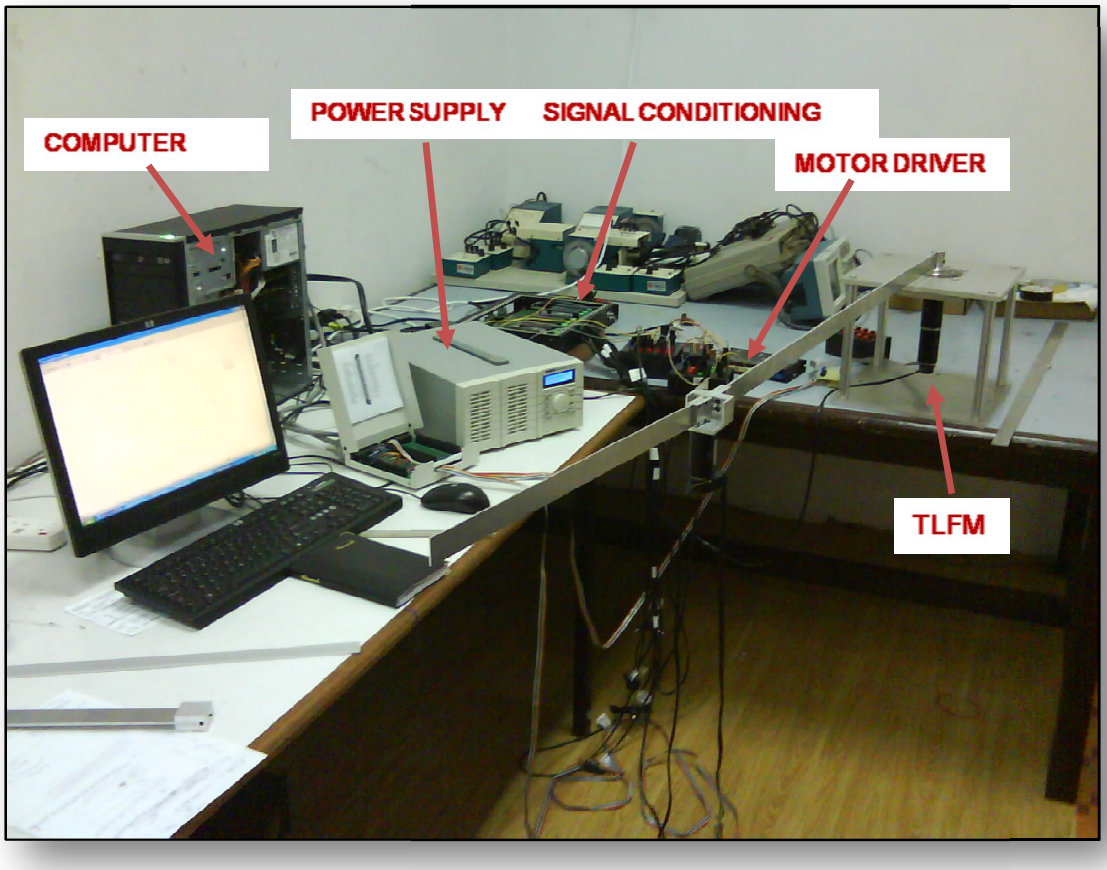


Figure 5.4: The actual image of hardware setup

Figure 5.3 shows the full description of TLFM system. The blue line is a digital output from computer while the red line is an analog output from encoders. The system has been simulated by using MATLAB/SIMULINK. The input for the system starts from the computer which gives the signal to the digital to analog converter (DAC) on Data Acquisition Card (DAQ). Then, the signal conditioning which is SCC 2345 has modified the signal and sent it to the motor driver. The signal conditioning will suit the signal to the motor driver. There are two DC motors that have been used to move the link one and link two. The motor driver 1 is responsible to control the first motor while the motor driver 2 is for controlling the second motor. The angular displacement of both motor will be measured by the encoder that has been integrated on both motor. The feedback signals from the encoders are connected to PCIQUAD04 for interfaced to the computer.

5.1.3 Encoder Installation and Calibration

Encoder is used for measuring the angular displacement of the DC-motor and it is integrated on the DC-motor shaft. The installation of the encoder needs to be done carefully in order to get the best reading of the motor position. Encoder calibration is intended to get an angular displacement from the pulse count by the optical encoder. Figure 5.5 shows the encoder installation on the motor shaft. After encoder installation on the motor shaft, the encoder has been connected to the PCI QUAD04 through the SCB 37 connection box. PCI QUAD04 functions as a counter and sent the angular displacement reading to the computer.

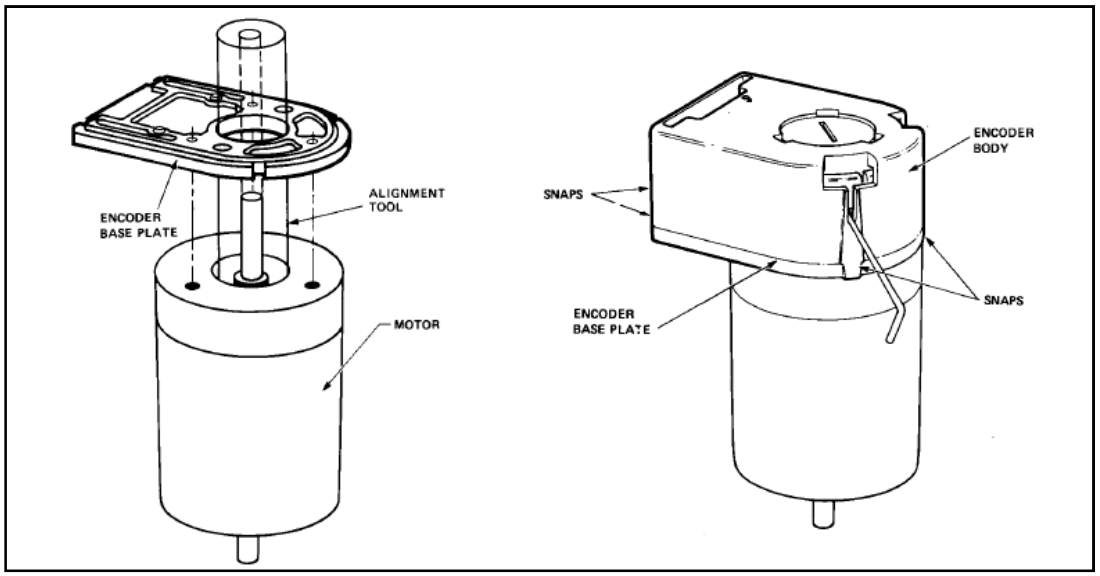


Figure 5.5: Encoder installation on the motor

For the encoder calibration, the encoder was tested by rotated the motor for one revolve. By using the Simulink block (scope), the result had showed the pulse counting by the optical encoder. The calibration steps are done for both motor. For the first motor, the gain that should be added on the Simulink block diagram is $\frac{2}{100\ \ 1000}$ while for the second motor the value is $\frac{}{1\ \ 0\ 1000}$. Both values are in radian.

5.2 Software Setup

MATLAB/SIMULINK is main software for this research project. This software provided with several toolboxes and the main toolbox that had been used in this project is a Real-Time Window Target (RTWT). RTWT toolbox is capable to run the system in a real-time. RTWT needs to be setup first in the computer before it can be used. The RTWT setup is briefly described in the next section.

5.2.1 Real-Time Window Target Setup

5.2.1.1 Installation and Configuration of RTWT

a) Installing the Kernel

During installation, all software for the Real-Time Windows Target is copied onto hard drive. The kernel is not automatically installed. The kernel can be installing just after RTWT has been installed. The installation of Kernel is needed before RTWT can be run. Next, is a steps to install the Kernel.

1. In the MATLAB window, type `rtwintgt -install`
   MATLAB displays the following message
   ```
   You are going to install the Real-Time Windows
   Target kernel.
   Do you want to proceed? [y] :
   ```
2. Continue installing the kernel. Type “y”.
   MATLAB installs the kernel and displays the message
   ```
   The Real-Time Windows Target kernel has been
   successfully installed.
   ```
   The computer has to be restart if a “`restart`” message being displayed.
3. To check that the kernel was correctly installed or not, `rtwho` is typed.
   MATLAB should display a message similar to,

```
Real-Time Windows Target version 2.5.0 (C) the
MathWorks, Inc. 1994-2003
MATLAB performance = 100.0%
Kernel time slice period = 1 ms
```

b) Testing the installation

The model rtvdp.mdl does not have any I/O blocks, so this model can be run regardless of the I/O boards in the computer. Running this model will test the installation by running Real-Time Workshop, the Real-Time Windows Target, and the Real-Time Windows Target kernel. After the Real-Time Windows Target kernel being installed, the entire installation can be tested by building and running a real-time application. The RTWT includes the model rtvdp.mdl, which already has the correct Real-Time Workshop options selected by users. The following steps are for testing the installation.

1. `rtvdp` is typed in the MATLAB window, the Simulink model rtvdp.mdl window opened as shown in Figure 5.6.

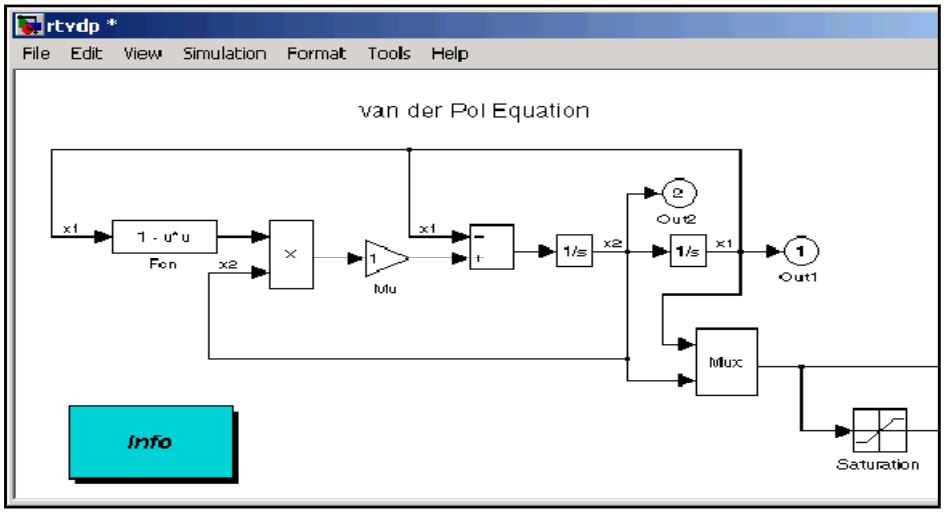


Figure 5.6: Simulink model rtvdp.mdl window

2. From the **Tools** menu, it should be pointed to **Real-Time Workshop** , and then click **Build Model** . The MATLAB window displays the following messages:

```
### Starting Real-Time Workshop build for model:rtvdp
### Invoking Target Language Compiler on rtvdp.rtw
. . .
### Compiling rtvdp.c
. . .
### Created Real-Time Windows Target modulertvdp.rwd.
### Successful completion of Real-Time Workshop build
procedure
For model: rtvdp
```

3. From the **Simulation** menu, click **External** , and then click **Connect to target**. The MATLAB window displays the following message:
   `Model rtvdp loaded`
4. **Start Real-Time Code is** clicked from **Simulation** menu.
5. The Scope window displays the output signals. After have been successfully installed the RTWT and run a real-time application, scope window should indicating such as shown in Figure 5.7.

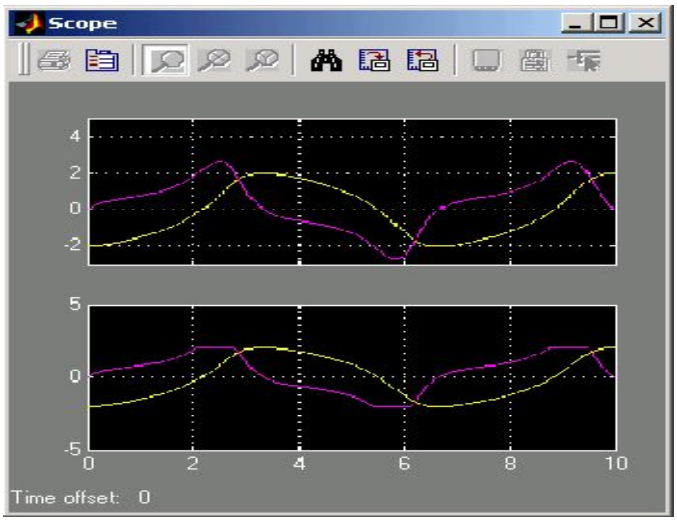


Figure 5.7: Scope Windows

6. From **Simulation** menu, click **Stop Real-Time Code**. The real-time application stops running, and the Scope window stops displaying the output signals.

5.2.1.2 Procedures of Creating Real Time Applications

a) Creating a Simulink Model

This procedure explains how to create a simple Simulink model. This model is used as an example to learn other procedures in the Real-Time Windows Target. A Simulink model need to be created before it can run a simulation or create a real-time application. The procedures to create a Simulink model are listed below.

1. In the MATLAB Command Window, type `simulink`. The Simulink Library Browser window is opened as shown in Figure 5.8.
2. From the toolbar, the **Create a new model** button is clicked.

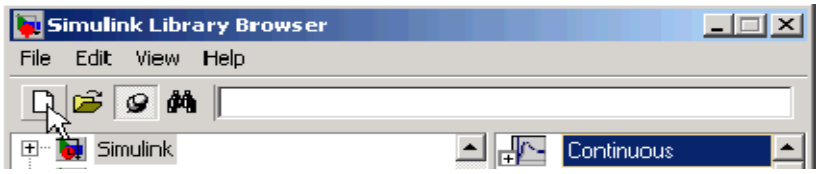


Figure 5.8: Created a new Simulink Model

An empty Simulink window is opened. With the toolbar and status bar disabled, the window looks like the Figure 5.9.

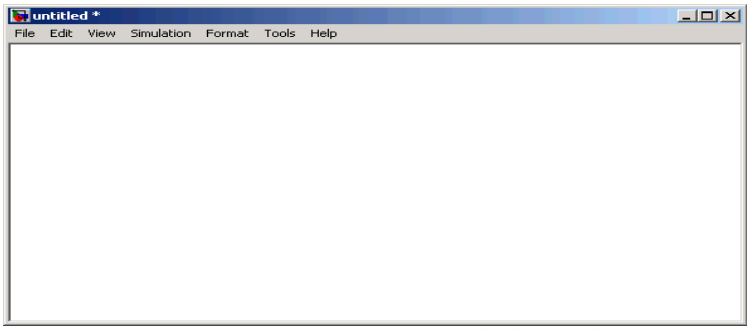


Figure 5.9: An empty Simulink window

3. In the Simulink Library Browser window, **Simulink** is double-clicked, and then **Sources** is also double-clicked. Next, **Signal Generator** is clicked and dragged to the Simulink window. Click **Continuous**. Click and drag **Transfer Fcn** to the Simulink window. Click **Sinks**. Click and drag **Scope** to the Simulink window.
4. The **Signal Generator** output is connected to the **Transfer Fcn** input by clicking-and-dragging a line between the blocks. Likewise, connect the **Transfer Fcn** output to the **Scope** input.
5. Double-click the **Transfer Fcn** block. The **Block Parameters** dialog box opens. In the **Numerator** text box, enter [10000]. In the **Denominator** text box, enter [1 70 10000]. **Block Parameters** dialog box will show look similar to the Figure 5.10 as shown below.

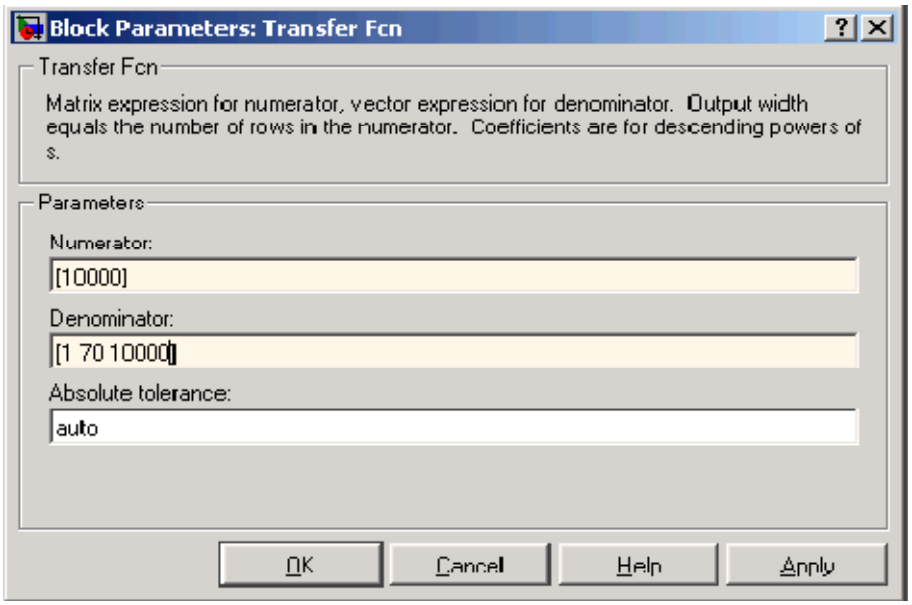


Figure 5.10: Block parameters dialog block

6. Click **OK**.
7. Double-click the Signal Generator block. The **Block Parameters** dialog box opens. From the **Wave form** list, select square. In the **Amplitude** text box, enter 1. In the **Frequency** text box, enter 20. From the **Units** lists, select rad/sec. Block parameters dialog box is shown in Figure 5.11.

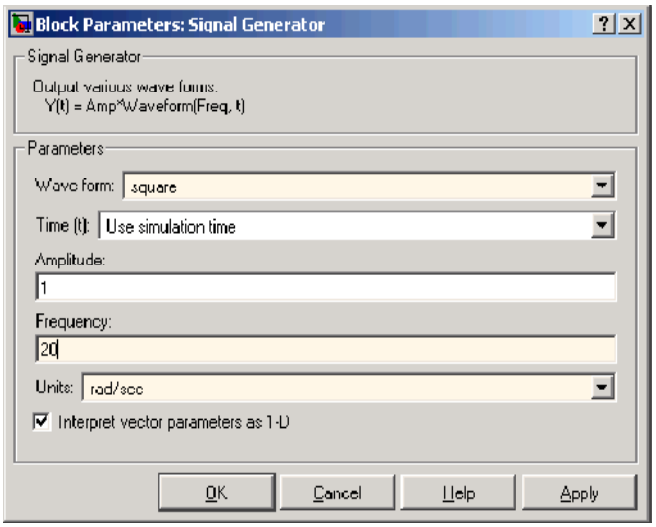


Figure 5.11: Block parameters of signal generator

After you add a Scope block to your Simulink model, you can enter the scope parameters for signal tracing: To configure a Real-Time Windows Target model to use an I/O board whether or not the board exists in the computer, but we will not be able to run the model until the board is installed with any jumpers or switches correctly set. Details of installation and configuration depend on the data transfer direction and the specific board, but are essentially similar in all cases. The following instructions use the National Instrument PCI6221 I/O board as an example, configure the board for analog input, and assume that you have physically configured and installed the board in your computer before you add its driver to your model. Customize the steps to provide the results that you need. Procedures below show how to install and configure an I/O board and its driver.

1. Install the board in the computer, setting any jumpers or switches as needed.
2. In the model window, choose View > Library Browser to display the Simulink Library Browser.
3. Clone an appropriate Input or Output driver block to your model from the Real Time Windows Target library.
4. Double-click the cloned I/O driver block.

The I/O Block Parameters dialog box opens. For an Analog Input block, the dialog box is shown in Figure 5.12.

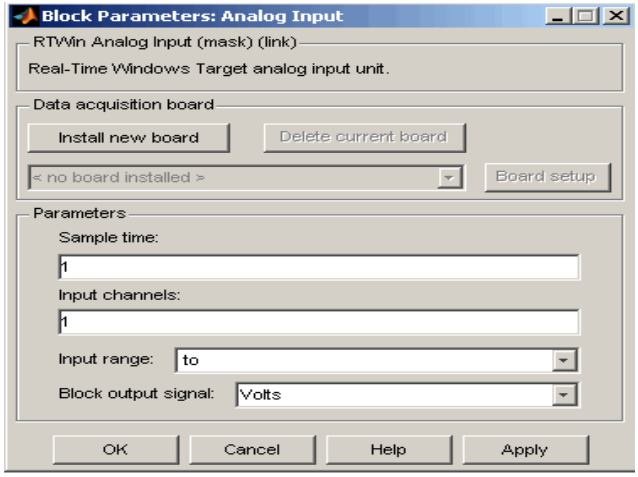


Figure 5.12: I/O Block Parameters dialog box

Click Install new board. From the list that appears, point to a manufacturer, then select a board type. For example, point to National Instrument, and then click PCI 6221. Then Test button is clicked.

The Real-Time Windows Target kernel tries to connect to the selected board, and if successful, displays the following message as shown in Figure 5.13 below.

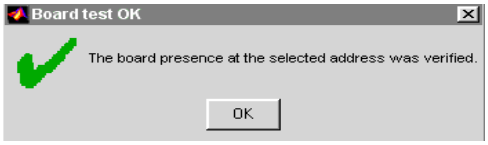


Figure 5.13: Display Message Dialog Box

b) PCI Bus Board

You do not have to set a base address with a PCI board. The plug-and-play feature of the Windows® operating system assigns a PCI slot number. You can enter the slot number in the I/O board dialog box, or you can let the driver determine the slot number for you. You open the I/O board dialog box from any I/O driver Block Parameters dialog box. As recommend that before we use a PCI or PCMCIA board, and then install the drivers supplied by the board manufacturer. The RTWT software does not use these manufacturer-supplied drivers. However, they sometimes initiate

the plug-and-play recognition of the board. Without these drivers installed, the board might be invisible to your computer and to the RTWT software.

RTWT applications cannot use Windows DLLs and kernel-mode drivers, which are not suitable for real-time operation. The device drivers supported by the RTWT software are listed at Supported I/O Boards. If no driver is listed for the board that you want to use, you may be able to write a custom device driver. A user-written custom device driver must program the board directly at the register level, which is the technique used by all supported RTWT drivers. If the board registers are I/O-mapped, the RTWT software supports such programming. If the registers are memory-mapped, programming them would require mapping their memory region to RTWT address space, which the RTWT software does not support for user-written drivers.

1. In the Simulink window, double-click the Scope block. A Scope window opens.
2. The **Parameters** button is clicked as shown in Figure 5.14.

Figure 5.14: Scope parameters Dialog Box.

   A **Scope parameters** dialog box opens.
3. The **General** tab is clicked. In the Number of axes box, enter the number of graphs you want in one Scope window. For example, enter 1 for a single graph. Do not select the floating scope check box. In the Time range box, enter the upper value for the time range. For example, enter 1 second. From the Tick labels list, choose all. From the Sampling list, choose Sample time and enter 0 in the text box. Entering 0 indicates that Simulink evaluates this block as a continuous time block. If you have discrete blocks in your model, enter the Fixed step size you entered in the Configuration Parameters dialog box. Your Scope parameters dialog box will look similar to the Figure 5.15 shown below.

Figure 5.15: Scope parameters

4. Do one of the following:

- **Apply** is clicked to apply the changes to your model and leave the dialog box open.
- **OK** is clicked to apply the changes to your model and close the **Scope parameters** dialog box.

5. In the Scope window, point to the *y*-axis shown in the Figure 5.16 below, and right-click.

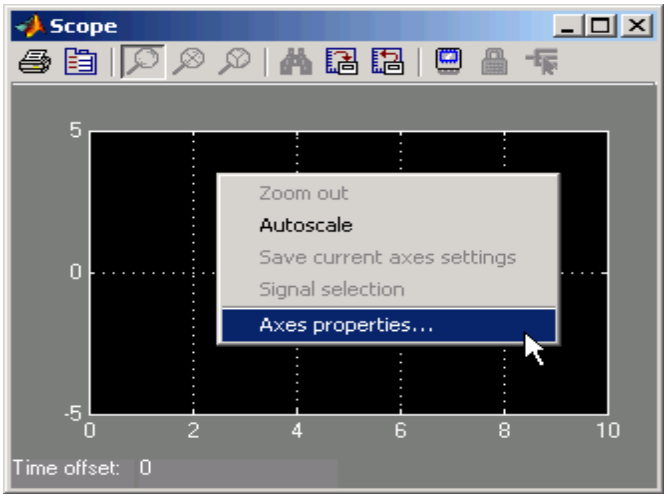


Figure 5.16: Scope window for *y*-axis Dialog Box

6. From the pop-up menu, click **Axes Properties**.
7. The **Scope properties: axis 1** dialog box opens. In the **Y-min** and **Y-max** text boxes, enter the range for the *y*-axis in the Scope window. For example, enter -2 and 2 as shown in the Figure 5.17 below.

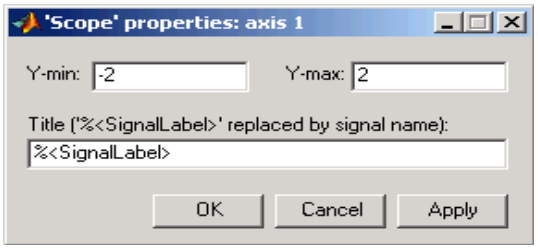


Figure 5.17: Scope properties: axis 1 Dialog Box

8. Do one of the following:

- **Apply** is clicked to apply the changes to your model and leave the dialog box open.
- **OK** is clicked to apply the changes to your model and close the **Axes Parameters.** The complete Simulink Block Diagram is shown in Figure 5.18.

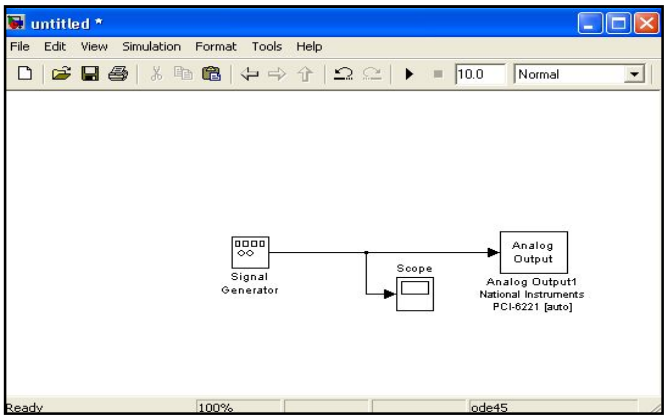


Figure 5.18: Completed Simulink Block Diagram

Save as is clicked from the **File** menu. In the File name text box, a **File name** for the Simulink model is entered and **Save** is clicked. For example, rtwin_model is typed Simulink saved the model in the file rtwin_model.mdl

c) Entering Configuration Parameters for Simulink

The configuration parameters give information to Simulink for running a simulation. After you create a Simulink model, you can enter the configuration parameters for Simulink. This procedure uses the Simulink model rtwin_model.mdl as an example and assumes you have already loaded that model.

1. In the Simulink window, and from the **Simulation** menu, click **Configuration Parameters.** In the **Configuration Parameters** dialog box, click the **Solver** tab. The **Solver** pane will open.
2. In the **Start time** box, enter 0.0. In the **Stop time** box, enter the amount of time you want your model to run. For example, enter 10.0 seconds.
3. From the **Type** list, choose Fixed-step. Real-Time Workshop does not support variable step solvers.
4. From the **Solver** list, choose a solver. For example, choose the general purpose solver ode5 (Dormand-Prince).
5. In the **Fixed step size** box, enter a sample time. For example, enter 0.001 seconds for a sample rate of 1000 samples/second.
6. From the **Tasking Mode** list, choose SingleTasking . For models with blocks that have different sample times, choose MultiTasking . Your **Solver** pane will look similar to the Figure 5.19 shown below.

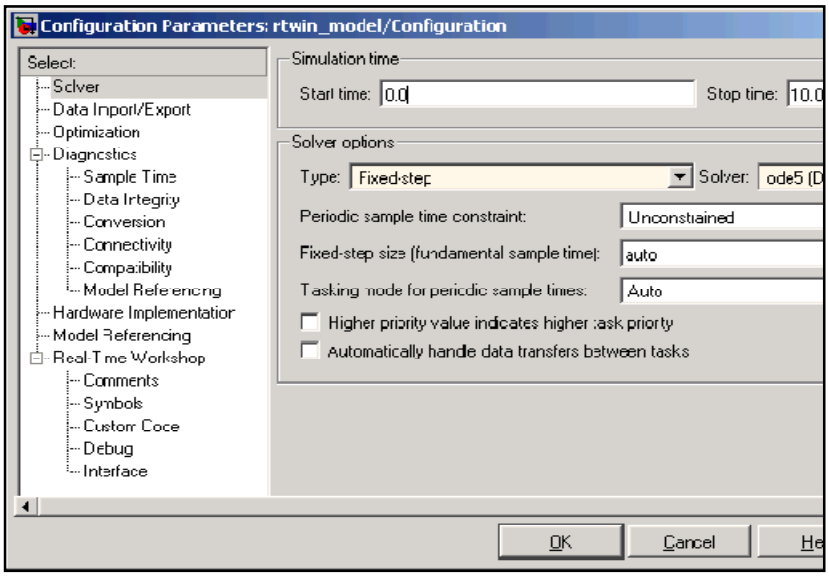


Figure 5.19: Solver pane Dialog Box

7. Do one of the following:

- **Apply** is clicked to apply the changes to your model and leave the dialog box open.
- **OK** is clicked to apply the changes to your model and close the **Configuration Parameters** dialog box.

d) Entering Simulation Parameters for Real-Time Workshop

The simulation parameters are used by Real-Time Workshop for generating C code and building a real-time application. The procedures for entering the simulation parameters are shown below.

1. In the Simulink window, and from the **Simulation** menu, click **Configuration Parameters**.
2. Click the **Hardware Implementation** node.
3. From the **Device type** list, 32-bit Real-Time Windows Target is chosen. Figure 5.20 shows the solver pane dialog box.

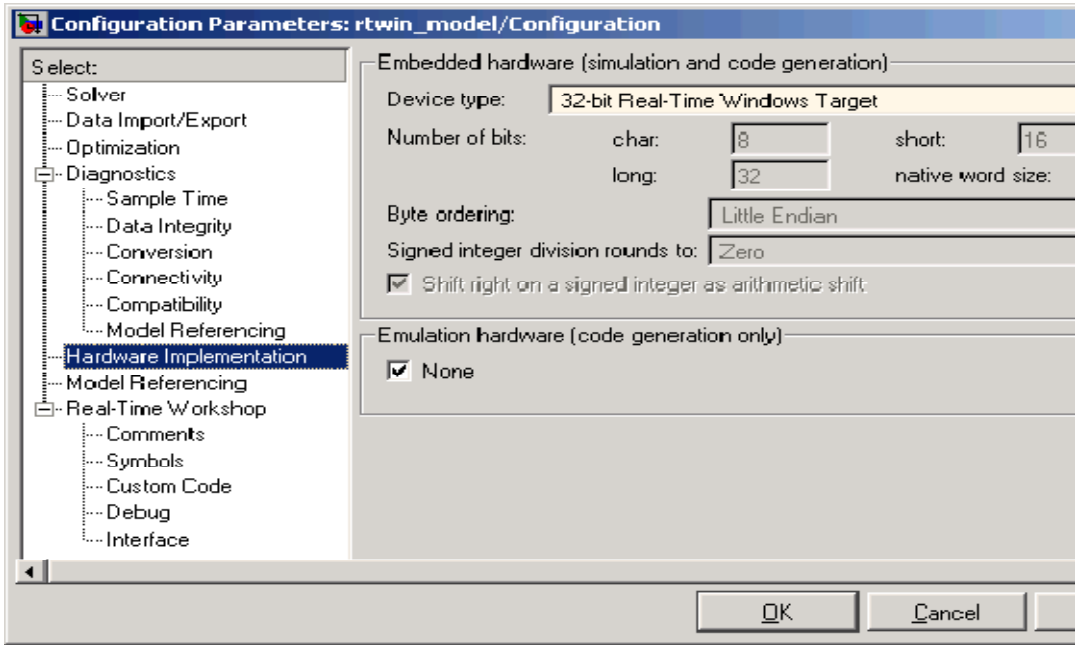


Figure 5.20: Solver pane Dialog Box for Hardware implementation

4. The **Real-Time Workshop** node is clicked. The **Real-Time Workshop** pane opens.
5. In the **Target selection** section, the **Browse** button is clicked at the **RTW system target file** list. The **System Target File Browser** opens as shown in Figure 5.21 below.
6. The system target file is selected for the Real-Time Windows Target and **OK** is clicked.

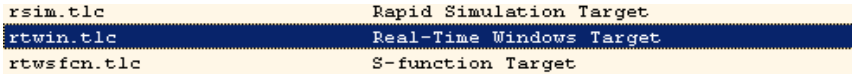


Figure 5.21: System Target File Browser

The system target file rtwin.tlc, the template makefile rtwin.tmf, and the make command make_rtw are automatically entered into the **Real-Time Workshop** pane. Although not visible in the **Real-Time Workshop** pane, the external target interface MEX file rtwin.ext is also configured after you click **OK**. This allows external mode to pass new parameters to the real-time application and to return signal data from the real-time application. The data is displayed in Scope blocks or saved with signal logging. Your **Real-Time Workshop** pane will look similar to the Figure 5.22 shown below.

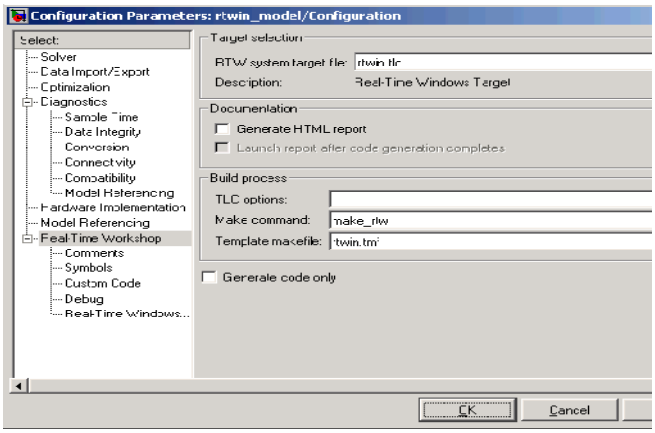


Figure 5.22: Real-Time Workshop pane

7. One of the following is done:

- **Apply** is clicked to apply the changes to your model and leave the dialog box open.
- **OK** is clicked to apply the changes to your model and close the **Configuration Parameters** dialog box.

e) Creating a Real-Time Application

Real-Time Workshop generates C code from your Simulink model, and then the Open Watcom C/C++ compiler compiles and links that C code into a real-time application. After you enter parameters into the **Configuration Parameters** dialog box for Real-Time Workshop, you can build a real-time application. This procedure uses the Simulink model rtwin_model.mdl as an example, and assumes you have loaded that model.

1. In the Simulink window, and from the **Tools** menu, point to **Real-Time Workshop**, and then click **Build Model**. The build process does the following:

- Real-Time Workshop creates the C code source files rtwin_model.c and rtwin_model.h.
- The make utility make_rtw.exe creates the makefile rtwin_model.mk from the template makefile rtwin.tmf.
- The make utility make_rtw.exe builds the real-time application rtwin_model.rwd using the makefile rtwin_model.mk created above. The file rtwin_model.rwd is a binary file that we refer to as your real-time application. You can run the real-time application with the RTWT kernel.

2. Simulink model is connected to the real-time application. After you create a real-time application, you can exit MATLAB, start MATLAB again, and then connect and run the executable without having to rebuild.

f) Running a Real-Time Application

The real-time application is run to observe the behavior of your model in real time with the generated code. The process of connecting consists of:

- Establishing a connection between Simulink model and the kernel to allow exchange of commands, parameters, and logged data.
- Running the application in real time.

After the real-time application is build, the model can be in real time. Below are the steps to run the Real-time application.

1. From the **Simulation** menu, click **External**, then from the **Simulation** menu click **Connect To Target** as illustrates in Figure 5.23**.** Also, you can connect to the target from the toolbar by clicking .

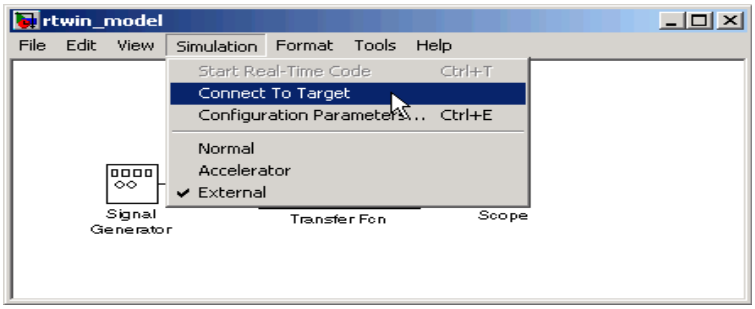


Figure 5.23: Connect to target from the simulation menu

MATLAB displays the message

```
Model rtwin_model loaded
```

2. In the Simulation window, and from the **Simulation** menu, **Start Real-Time Code** is clicked as shown in Figure 5.24. The execution also could be start from the toolbar by clicking .

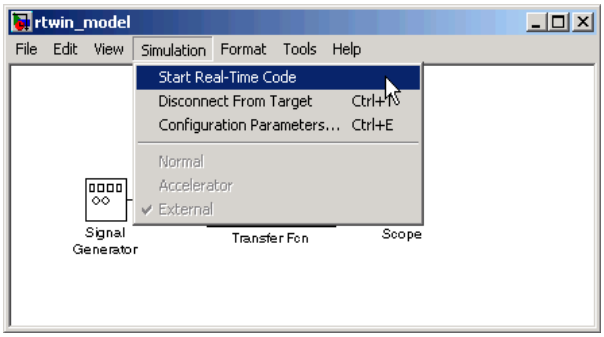


Figure 5.24: Start Real-Time Code from simulation menu

Simulink runs the execution and plots the signal data in the Scope window. In this example, the Scope window displays 1000 samples in 1 second, increases the time offset, and then displays the samples for the next 1 second.

3. Do one of the following:

- Let the execution run until it reaches the stop time.
- **Stop Real-time Code** is clicked from the **Simulation** menu. The real-time application is stopped.

4. In the Simulation window, and from the **Simulation** menu, click **Disconnect from Target**.
5. From the **Simulation** menu, **External** is clicked**.**

   MATLAB displays the message

   `Model rtwin_model unloaded`

5.2.2 Simulink Block Diagram for TLFM

After RTWT setup was done, the Simulink block diagram was created to control the TLFM. The Simulink block diagram for controlling the TLFM is illustrates in Figure 5.25.

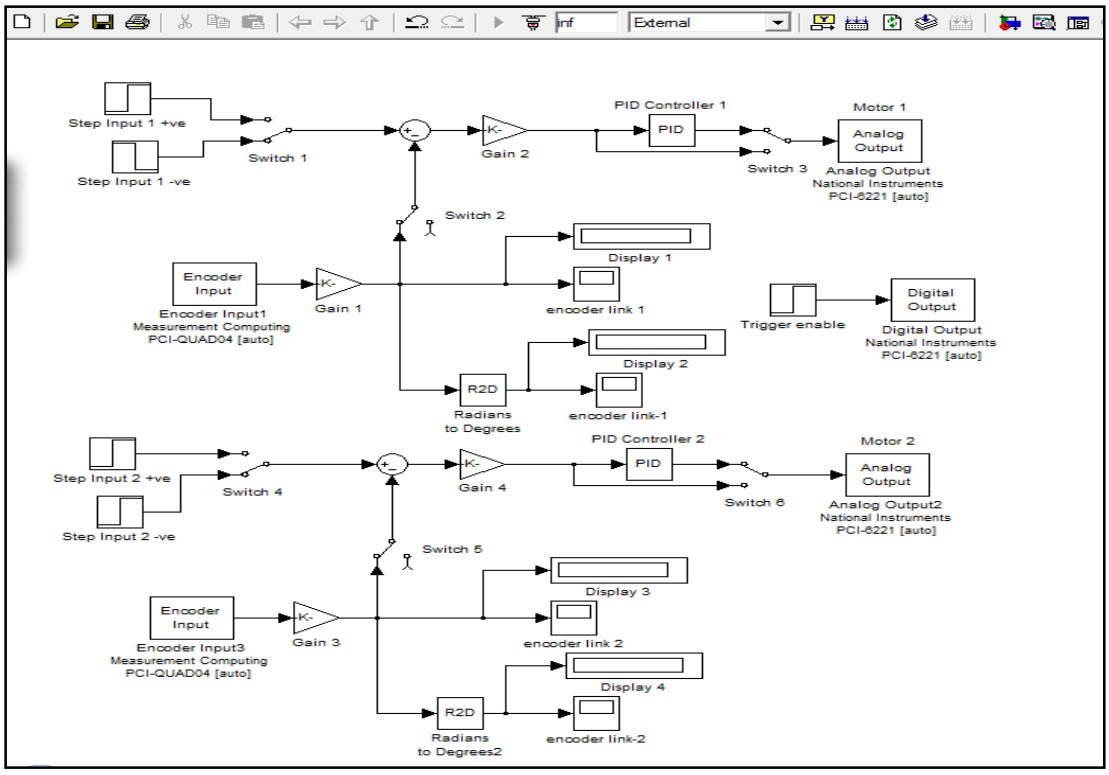


Figure 5.25: The Simulink block diagram for TLFM

The Simulink block diagram above can be divided into two parts; the upper part is for controlling the first motor while the below part is for controlling the second motor. Trigger enable input (+5V) is for enabling the both motor drivers. The trigger enable is connected to digital output block which is physically is a PCI 6221 digital output. Physically, the digital output line from SCC 2345 was connected to the both motor driver's enable line in parallel. That's mean the both motor driver will switch on simultaneously. Meanwhile, the operation of both motor is depends on the input that been supplied. Step input was supplied to both motor and it consists of positive and negative input. Switch 1 and switch 4 is used to change the positive and negative input for first motor and second motor respectively. Encoder input 1 and encoder input 2 are for read the encoder signal from first motor and second motor respectively. The response of encoders can be viewed with scope block. Physically, both encoders were connected to PCI QUAD04 through connection box SCB 37 where channel 4 is for first encoder and channel 1 is for second encoder. Gain 1 and gain 3 is for encoder's calibration value for first motor and second motor. The value for gain 1 is $\frac{2}{100\ \ 1000}$ and the value for gain 3 is $\frac{}{1\ \ 0\ 1000}$. Thus, the readings of the angular displacement for both motor are in radian. To convert the angular displacement in degree, radians to degrees block has been used.

To control the motors, it depends on the input to the set values on motor driver. The analog output 1 and analog output 2 blocks are for the set value to motor driver 1 and motor driver 2 respectively. The analog output is on the PCI 6221 and connected to SCC 2345. Channel 1 analog output is for first motor while channel 2 analog output is for second motor. In controlling the TLFM, two PID controllers have been used. Each of them is for each motor. The value of Kp, Ki and Kd were tuned to get the best response of the TLFM. The PID tuning methods are discussed in the next section. Switch 2 and switch 5 are responsible to change the operation of TLFM, whether in open loop or closed loop manner while switch 3 and switch 6 are responsible to switch between operation with PID controller or without PID controller.

5.3 PID Tuning

In this project, Ziegler-Nichols closed loop method is used for tuning the PID. It is performed by setting the *I* and *D* gains to zero. The "P" gain is then increased (from zero) until it reaches the **ultimate gain** *Ku*, at which the output of the control loop oscillates with a constant amplitude as shown in Figure 5.26. *Ku* and the oscillation period *Pu* are used to set the P, I, and D gains depending on the type of controller used.

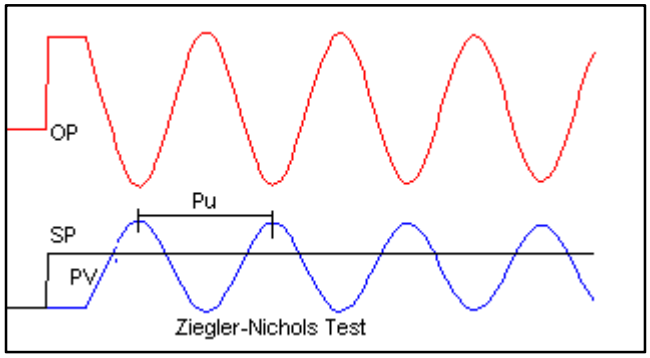


Figure 5.26: Stable oscillation

The ***closed loop method*** prescribes the following procedure:

Step 1: Disable any D and I action of the controller (--> pure P-controller)
Step 2: Make a setpoint step test and observe the response
Step 3: Repeat the SP test with increased / decreased controller gain until a stable oscillation is achieved. This gain is called the "ultimate gain" Ku.
Step 4: Read the oscillation period Pu.
Step 5: Calculate the parameters according to the following formulas:

Tuning Equations,

- P Only: Gain = 0.5Ku
- PI: Gain=0.45Ku, Reset=1.2/Pu
- PID: Gain=0.6Ku, Reset=2/Pu, Derivative=Pu/8

Where; K=Gain
R=Reset value
T=derivative

Then, from format of PID controller Gc= K(1 +R/s +T*s), the value of Kp = K, Ki = K*R and Kd = K*T.

After doing some experiments and calculations, the initial PID controller parameters are obtained for first motor and second motor are listed in Table 5.1.

Table 5.1: The initial PID controller parameters for DC-motors

DC-motor	Kp	Ki	Kd
First motor	0.060	0.2553	0.0035
Second motor	0.060	0.1538	0.0059

CHAPTER 6

RESULT AND DISCUSSION

One of the objectives in this project is to control the TLFM by using a real-time controller. Several experiments and control approach have been done to TLFM system. The result is clarified in this chapter. The first technique that has been done to TLFM system is open loop system which is without the feedback from the encoder. The second technique is closed loop system with the feedback signal from encoder but without the PID controller. And the last but not least technique is closed loop system with PID controller. All the experiments are done with first motor and second motor are operates simultaneously.

6.1 Open Loop (Feed forward) System

The open loop system is a system without a feedback from the encoder. The input for this system is a step input with the value of 1 radian for first motor and 0.4 radians for second motor. Positive step input and negative step input are supplied which the motors are moved in anticlockwise rotation and clockwise rotation respectively. Both motor were started at the same time. Figure 6.1 shows the result of the angular displacement of first motor with positive input, Figure 6.2 for first motor with negative input, Figure 6.3 for second motor with positive input and Figure 6.4 for second motor with negative input.

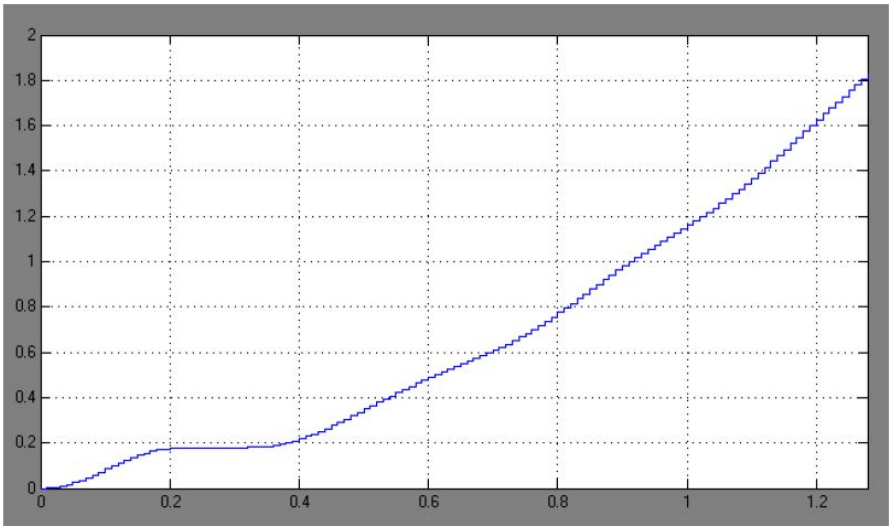


Figure 6.1: The angular displacement of first motor with positive input (open loop)

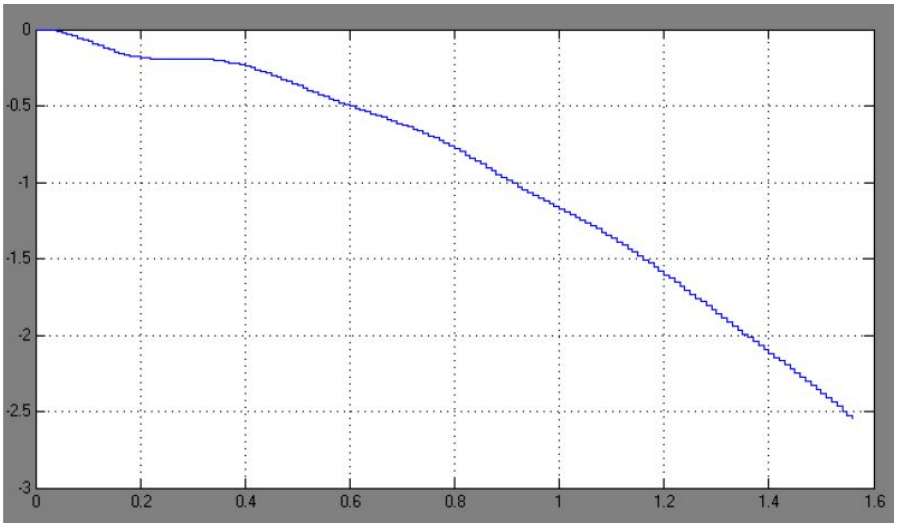


Figure 6.2: The angular displacement of first motor with negative input (open loop)

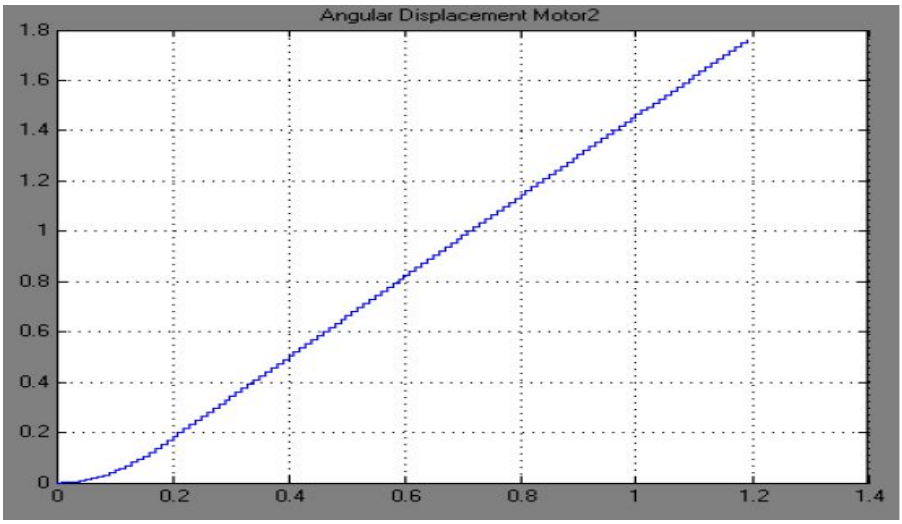


Figure 6.3:The angular displacement of second motor with positive input(open loop)

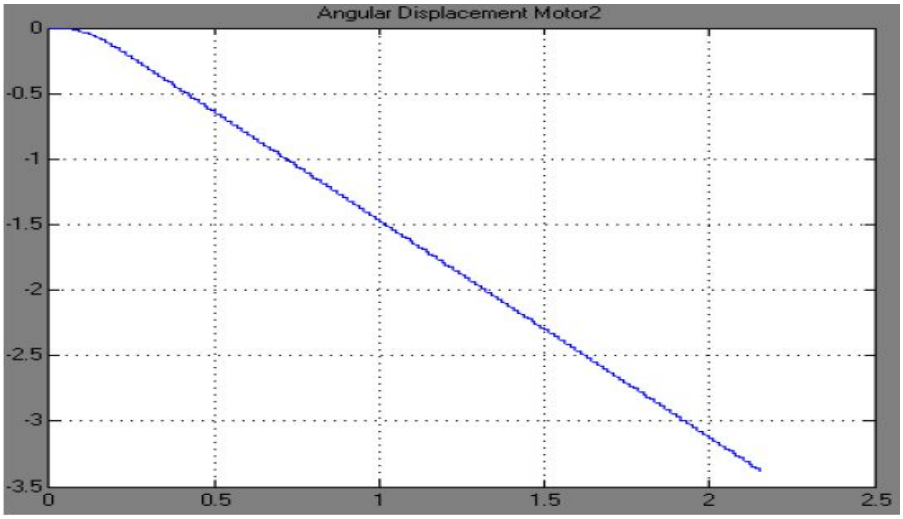


Figure 6.4:The angular displacement of second motor with negative input(openloop)

From all the results above, both motor are just run continuously until it stop manually. Those results are to show that without any feedback signal, the output of the angular displacement cannot follow the desired input. That's mean, the first motor are not stopped at 1 radian and the second motor unable to stop at 0.4 radians. But, from the open loop results, the angular velocity can be obtained by calculated the slope of the angular displacement response. For the first motor with positive input, the angular velocity is 2 rad/s while for negative input; the angular velocity is 2.67 rad/s. For the second motor with positive input, the angular velocity is 1.6 rad/s while for negative input; the angular velocity also 1.6 rad/s.

6.2 The Closed Loop (Feedback) System without PID Controller

The closed loop system is a system with a feedback from the encoder. The error (the difference between the output and the desired input) is feedback to influence the control action and improve the overall performance. In this section, the result of the closed loop system without the PID controller is discussed. Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8 show the result of angular displacement of first motor with positive rotation, first motor with negative rotation, second motor with positive rotation and second motor with negative rotation respectively.

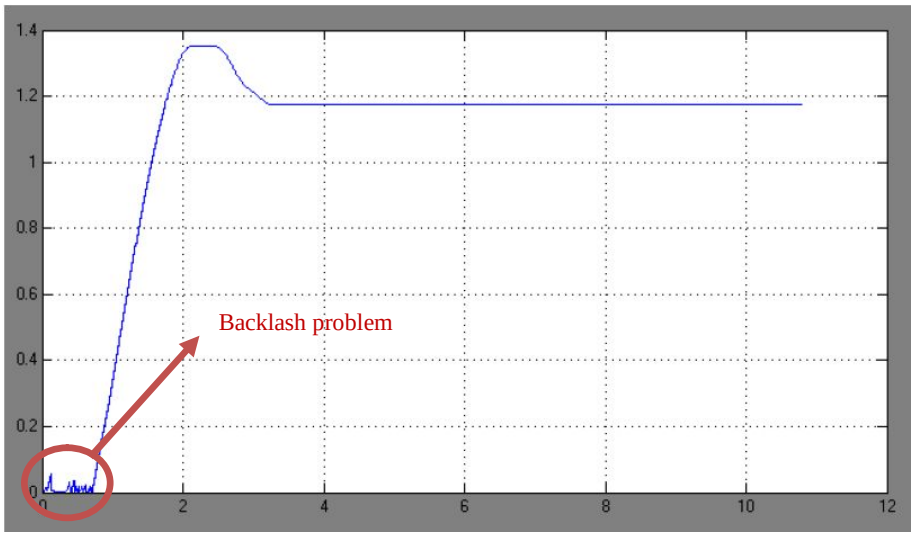


Figure 6.5: The angular displacement of first motor with positive rotation (closed loop)

Figure 6.5 above shows the result of angular displacement of first motor with positive rotation. According to the result, the response has an overshoot about 15.52% and the steady state error about 0.16 radians. The settling time, T_s is about 3.2 seconds while for the time constant, T_c is equal to 1.1 seconds. Unfortunately, there is a backlash problem to the motor. Backlash is the error in motion that occurs when gears change direction. It exists because there is always some gap between the trailing face of the driving tooth and the leading face of the tooth behind it on the driven gear, and that gap must be closed before force can be transferred in the new direction. This problem leads to a ripple at initial state and cause some delay about 0.8 seconds.

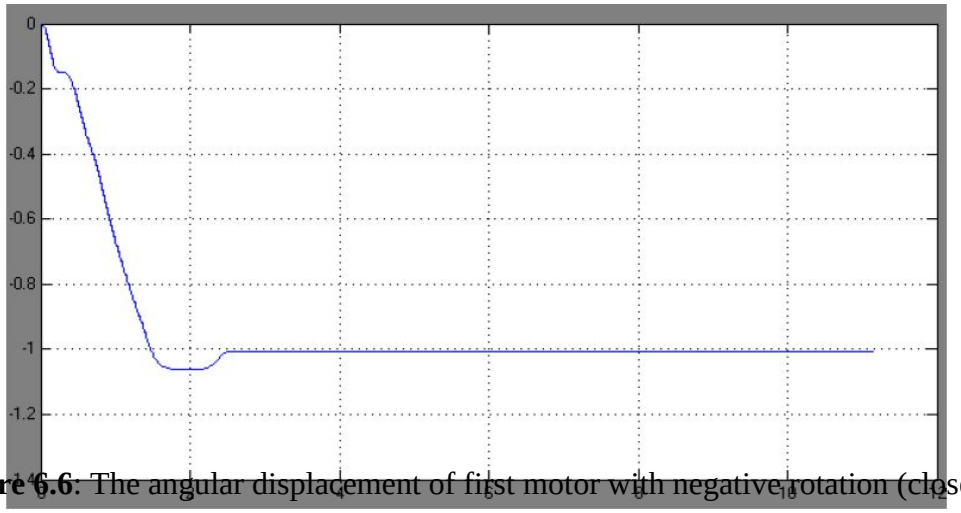


Figure 6.6: The angular displacement of first motor with negative rotation (closed loop)

From the result shown in Figure 6.6, the angular displacement of first motor with negative rotation has an overshoot about 3.92% and steady state error about 0.02 radians. The settling time, T_s and the time constant, T_c is about 2.45 seconds and 1.18 seconds respectively. There is no backlash problem in negative rotation, therefore, no delay at initial state.

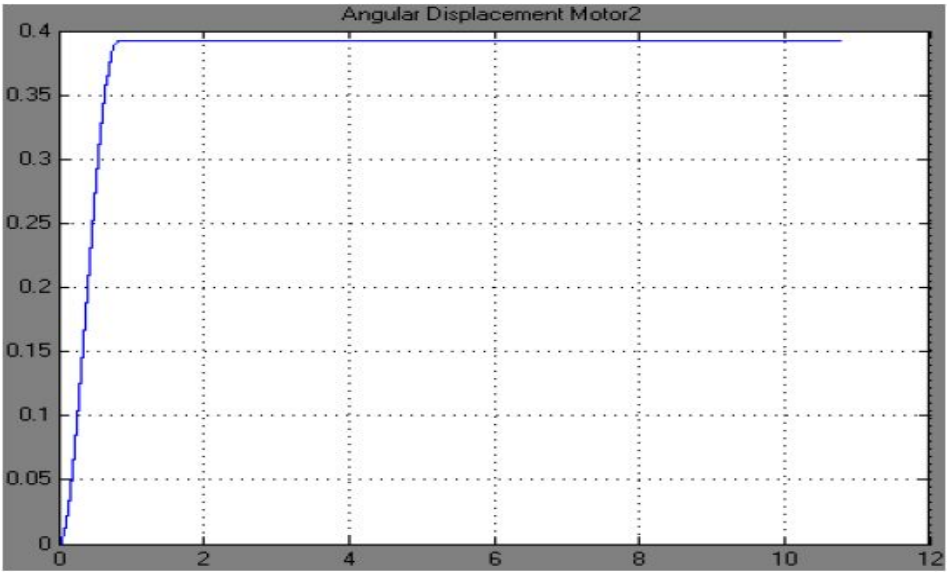


Figure 6.7: The angular displacement of second motor with positive rotation (closed loop)

Figure 6.7 shows the angular displacement of second motor with positive rotation. From the result above, the response is quite good but there is a steady state error about 0.02 radians. The settling time, T_s is 0.86 seconds while the time constant is 0.58 seconds. There are no overshoot and delay.

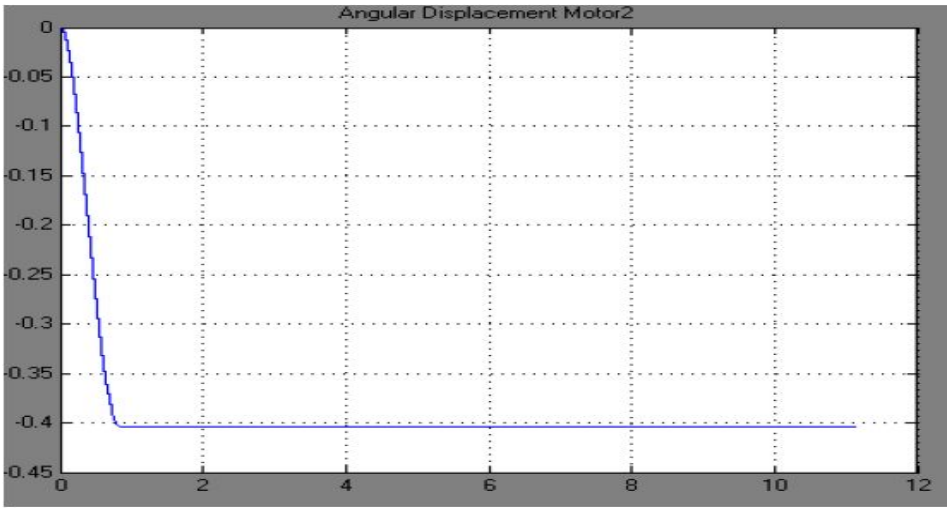


Figure 6.8: The angular displacement of second motor with negative rotation (closed loop)

Figure 6.8 shows the result of second motor with negative input. The result also quite good compared to first motor. There are no overshoot and no backlash problem. But, there is a steady state error about 0.006 radians. The settling time, T_s is equal to 0.86 seconds and T_c is equal to 0.44 seconds.

6.3 The Closed Loop System with PID Controller

The PID controller is easy to implement and robust stability. Therefore, this controller is a first choice to control the TLFM system. In this section, the result of a closed loop system with PID controller is discussed. The PID tuning is determined by using Ziegler-Nichols method. Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12 show the results of the angular displacement for first motor with positive input, first motor with negative input, second motor with positive input and second motor with negative input respectively.

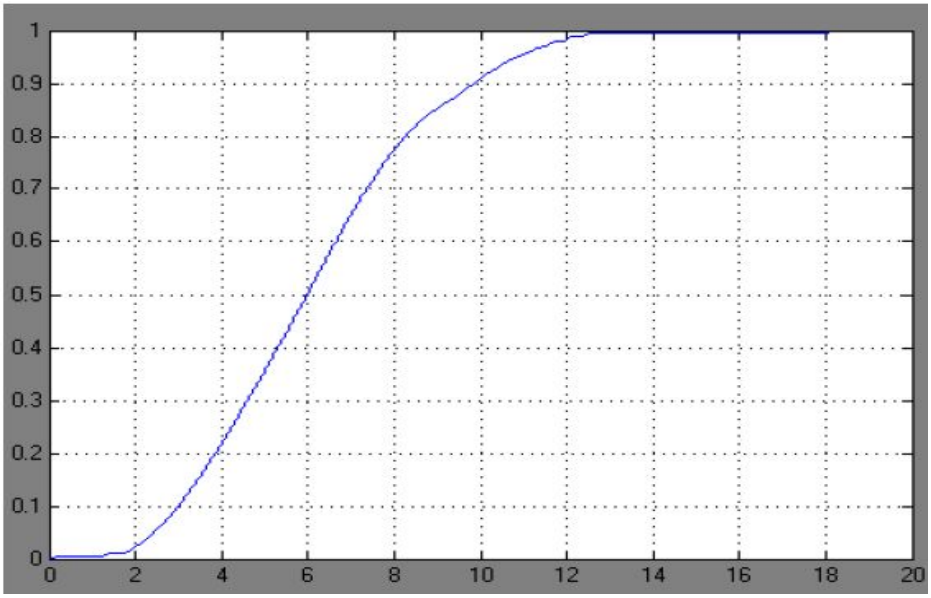


Figure 6.9: The angular displacement of first motor with positive input (closed loop with PID)

The value of PID parameters are Kp = 0.013, Ki = 0.004 and Kd = 0.05. From the result above, the settling time, T_s and the time constant, T_c is equal to 12.86 seconds and 7 seconds respectively. Although the time to reach the steady state is much longer compared to closed loop without PID controller system, the backlash

problem has been eliminated. Therefore, there is no delay and ripple at the initial state. That's mean the first motor was operates smoothly at the beginning. Furthermore, the overshoot problem also has been eliminated and there is no steady state error at all. Thus, the first motor was stopped at the desired output that is 1 radian (57.3 degrees).

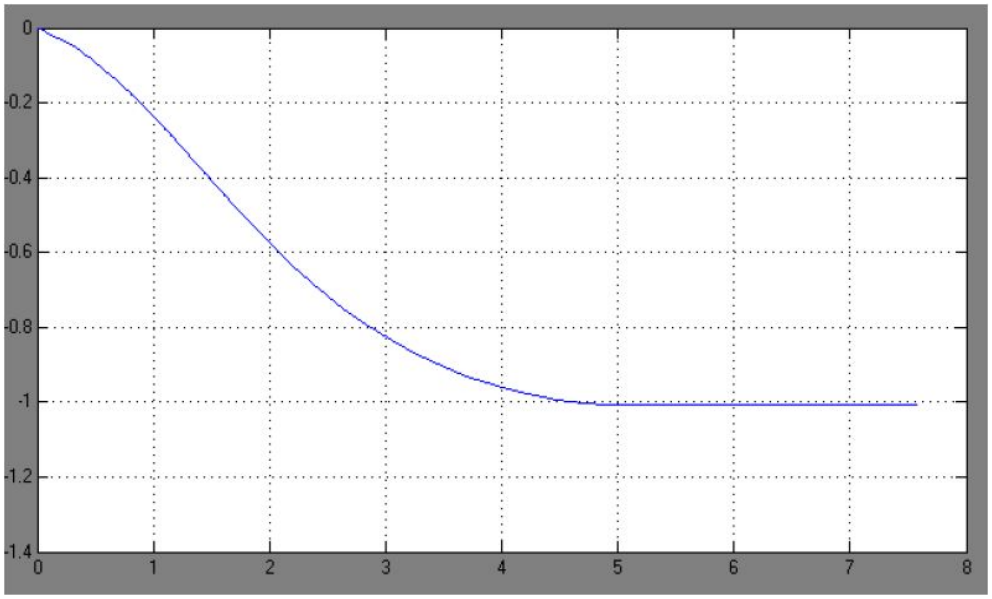


Figure 6.10: The angular displacement of first motor with negative input (closed loop with PID)

The value of PID parameters for this experiment are Kp = 0.0444, Ki = 0.0016 and Kd = 0.05. From the Figure 6.10, the result of first motor with negative input is better than positive input. The settling time and time constant is much faster compared to positive rotation. The T_s and T_c is about 4.78 seconds and 2.3 seconds respectively. Although the time to reach the steady state condition is longer compared to closed loop system without PID controller, the overshoot and steady state error have been eliminated. From the response of angular displacement, we can see that the first motor was rotating smoothly and stopped at desired output.

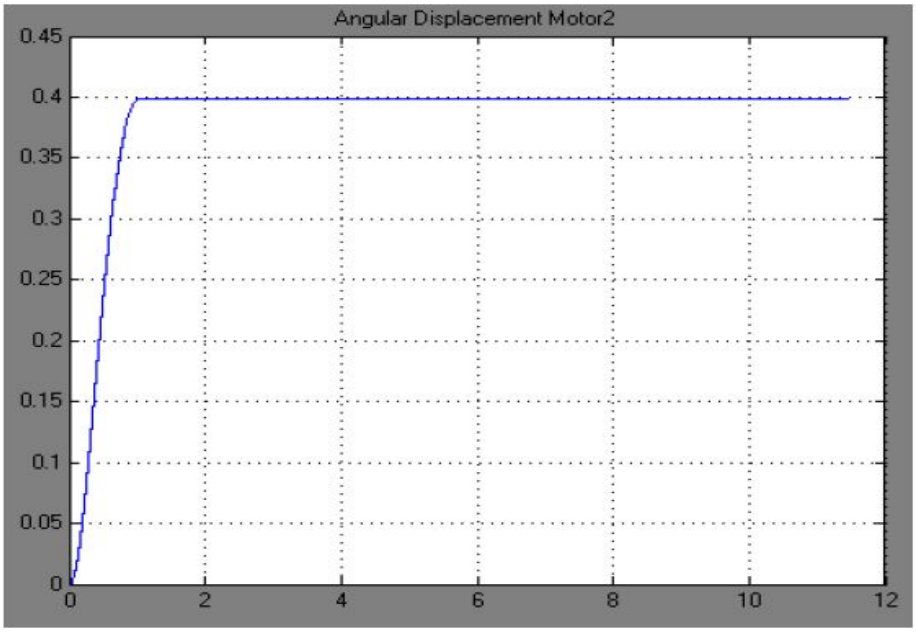


Figure 6.11: The angular displacement of second motor with positive input (closed loop with PID)

The value of PID parameters are Kp = 0.06, Ki = 0.03665 and Kd = 0.01. The response of angular displacement for second motor above shows that there is no steady state error that's mean the second motor stopped at 0.4 radians (22.92 degrees). The T_s is longer than the closed loop without PID controller which is 1 second but it has better time constant which is faster 0.01 seconds compared to without PID controller.

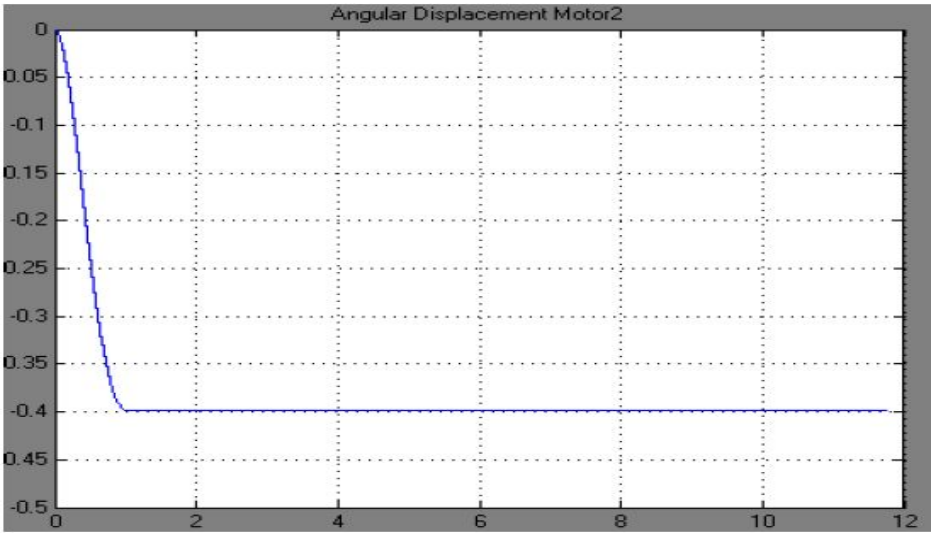


Figure 6.12: The angular displacement of second motor with negative input (closed loop with PID)

The value of PID parameters are Kp = 0.06, Ki = 0.0325 and Kd = 0.01. From the Figure 6.12, the response of second motor is approximately the same with response without PID controller. The settling time is longer than without PID controller that is about 1 second. But, with PID controller the time constant is reduced to 0.43 seconds which is 0.01 seconds faster than without PID system. Furthermore, the steady state error has been eliminated due to the sense of PID controller and certainly the second motor was stopped at 0.4 radians.

6.4 Discussion

The problem occurred in the first motor is a backlash problem of the gear. That problem only happened in the positive (anti-clockwise) rotation but not in the negative (clockwise) rotation. This trouble shows that the first motor has a problem with a gear when it rotated in anti-clockwise. But, this problem has been solved by PID controller system although it takes longer time to reach the steady state condition. For the second motor, the responses are approximately the same between closed loop without PID controller system and closed loop with PID controller system. The PID controller just eliminated the steady state error. The system without PID controller has no overshoot because the second motor just hold the small load compared to first link that has to hold the weightier load. That’s why the response of first link has an overshoot when it operates without PID controller.

PID controller parameters have a different value for the different rotation of first and second motor. Therefore, the more PID tuning is needed. This condition maybe because of the DC-motor has a different friction for the different rotation. For example, the first motor has a less friction when it runs in clockwise rotation. It has been proved from the result of closed loop without PID controller where in a clockwise rotation, there is no backlash problem. Also, by referring to open loop system, the angular velocity of negative rotation is faster than the positive rotation.

Overall, the PID controller has improved the response of TLFM. For first motor with positive rotation, the PID controller has eliminated the backlash problem of the gear and also no steady state error. The overshoot also has been removed and no delay at the initial state. In negative rotation, the overshoot and steady state error have been removed at all. Meanwhile, for the second motor with positive and negative rotation, the steady state error has been eliminated. The result of controlling the TLFM is summarized in Table 6.1.

Table 6.1: Experimental Result

System		Overshoot (%)	Backlash	Td (s)	SSE (rad)	Ts	Tc
No PID	M1+ve	15.52%	yes	0.8	0.16	3.2	1.1
	M1-ve	3.92	no	0	0.02	2.45	1.18
	M2+ve	0	no	0	0.02	0.86	0.58
	M2-ve	0	no	0	0.006	0.86	0.48
With PID	M1+ve	0	no	0	0	12.86	7
	M1-ve	0	no	0	0	4.78	2.3
	M2+ve	0	no	0	0	1	0.57
	M2-ve	0	no	0	0	1	0.43

where, M1 = first motor

M2 = second motor

Td = time delay

SSE = steady state error

Ts = time settling

Tc = time constant

+ve = positive rotation

-ve = negative rotation

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

As a conclusion, the TLFM has been developed and controlled by using PID controller. At the early stage, the mechanical structure has been designed and then it has been fabricated by using the aluminum. The mechanical structure is tidy and robust in order to implement the TLFM system properly. The interfaced between hardware and software also have been done for controlling the TLFM by using MATLAB/SIMULINK. The signal from encoder has been studied to implement it as a feedback signal for control of TLFM system. After doing several experiment setups of hardware and software, three experiments have been done towards TLFM system which is open loop system, closed loop system and closed loop with PID controller system. PID controller has been developed and implemented on the TLFM system and from the results obtained from those experiments; the system with PID controller has improved the performance of TLFM operation. Thus, the TLFM is operated smoothly and move with accurate position by the control of PID controller.

7.2 Recommendations

Despite the TLFM system has been successfully developed and controlled by using PID controller, there is still lack in some aspects. The TLFM system cannot measure the deflection of the links. Therefore, it is unable to give a feedback signal from the link deflection to the controller in order to reduce and eliminate the vibration

of the link. In spite of having a smooth movement of TLFM system, there are still have a small vibration or oscillation at the links because this system just has a feedback signal from the encoder integrated on a DC-motor. In order to overcome this problem, the direct strain feedback can be added to the system. The strain or deflection of links can be measured by using a strain gauge. The strain gauges can be placed on the link at several locations to measure the strain of the link uniformly. Besides that, the accelerometer can be used to measure the end-point displacement and acceleration of the link. Furthermore, the accelerometer signal can be used as a feedback for developing the optimum controller.

As we can see in the experiment result for PID controller, the parameters for PID controller is different for any different movement. PID controller are fixed and specified by the designer. So, the PID with constant Kp, Ki and Kd cannot cope with the changes of motor rotation or changes in plant parameters. This may not sufficient to provide the satisfactory performance for TLFM system. Therefore, the adaptive and self tuning controller is suitable to replace the PID controller. The adaptive controller is a type of controller that has the ability to adjust itself to any parameter variations occurring in a control system. As a result, the TLFM system can be improved because there is no need to change the PID constants for any changes in motor rotation. In addition, the Fuzzy Logic controller or Neural Network controller can be applied to the TLFM system for advance control.

The main problem to the first DC-motor is a backlash problem. This happened because of the problem with the gear. Hence, the high-end DC-motor and without backlash problem can be used so that the TLFM can moved faster and properly. Besides that, the feed forward control such as input shaping and filtering techniques can be implemented in this system. Moreover, the payloads with different weight can be attached at the end-point of second link. Then, the more experiments and control approaches can be done by using the different weight of loads.

REFERENCES

A. Mahmood, B. Bhikkaji and S. O. R. Moheimani (2007). *Vibration and Position Control of a Flexible Manipulator*. School of Electrical Engineering and Computer Science, the University of Newcastle Callaghan, NSW, Australia.

Ahmad Suryo Arifin, Edi Leksono, Hilwadi Hindersah, Wahyudi Martono and Riza Muhida (2007). *Two degree of freedom flexible link control using Fuzzy tuned PID controller.* Engineering Physics Study Program, Institut teknologi Bandung, June 2007.

Dwivedy, S. and Eberhard, P (2006). Dynamic analysis of flexible manipulators, a literature review. *Mechanism and Machine Theory, Vol. 41, 2006, pp. 749–777.*

Gulay Oke and Yorgo Istefarwpulos (2001). Gradient-Descent Based Trajectory Planning for Regulation of a Two-Link Flexible Robotic Arm. *IEEHASME International Conference on Advanced Intelligent Mechatronics Proceedings.* 8-12 July 2001. Como, Italy.

Jan-Peter Houschild (2003). *Control of a Flexible Link Robotic Manipulator in Zero Gravity condition.* University of Waterloo: A report in fulfilment of the requirement of GENE 503, August 2003.

Jinjun Shan, Hong-Tao Liu, Dong Sun (2004). Modified input shaping for a rotating single link flexible manipulator. *Journal of Sound and Vibration*. 2004. Vol. 285: 187-207.

Ke-Ping Liu (2007). Experiment Evaluation of Preshaped Input to Reduce Vibration for Flexible Manipulator. *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics.* 19-22 August 2007. Hong Kong: 2411-2415.

M.A Ahmad (2006). *Control of Flexible Manipulator*. Bachelor Degree. Faculty of Electrical Engineering, Universiti Teknologi Malaysia.

M. Benosman, G.L. Vey (2004). Control of flexible manipulators: *A survey, Robotica 22* (2004): 533–545.

Mazlan Muhammad (2009). *Real-Time Control of a Single Link Flexible Manipulator*. Bachelor Degree, Faculty of Electrical Engineering, Universiti Teknologi Malaysia.

Mohamed,Z and Tokhi, M.O (2002). Vibration control of a single link flexible manipulator using command shaping techniques. *Proceeding of IMechE-I: Journal of System and Control Engineering*. 2002. Vol. 216(2): 191-210.

Mohd Asyraf Ahmad, Zaharuddin Mohamed (2007). Hybrid Control Schemes of a flexible robot manipulator. *The 2nd National Intelligent Systems And Information Technology Symposium (ISITS'07)*. Oct 30-31, 2007. ITMA -UPM, Malaysia.

Noraide Md Yusop (2009). *Control of Flexible Manipulator using Strain Feedback.* Bachelor Degree, Faculty of Electrical Engineering, Universiti Teknologi Malaysia.

Ram Somaraju (2003). *Modelling and Control of a single Flexible Beam*. The University of Auckland, Department of Electrical and Electronic Engineering, Auckland: Technical report.

Sungsoo Rhim and Ai-Ping Hu (2003). Control of a Flexible Manipulator Tip Using a Multirate Repetitive Learning Controller Based on Filtered Tip Reference Input. *Proceedings American Control Conference.* June 4-6. Denver Colorodo.

Victor Etxebarria, Arantza Sanz & Ibone Lizarraga (2005). Control of a Lightweight Flexible Robotic Arm Using Sliding Modes. *International Journal of Advanced Robotic Systems.* 2005. Vol 2(2): 103-110.

W.S. Jang, K.S. Kim and S.K. Lee (2003). An Open-Loop Control Scheme for Minimization of Residual Vibrations of a Flexible Robot. *2003 Society for Experimental Mechanics Experimental Mechanics:* 387.

Y. Yang, W.G.Wang (2002). A Fuzzy Parameter Adaptive PID Controller Design of Digital Positional Servo System. *Proceeding of the first International Conference on Machine Learning and Cybernetics. 2002:* 310-314.

Ying Sun, Runzhi Wu and Fangxiao Cheng (2007). Design of a Flexible Manipulator controller Based on Input-output Linearization. *Proceedings of the IEEE International Conference on Automation and Logistics*. August 18 – 21 2007. Jinan,China.

APPENDIX A
PCI-QUAD04

Specifications

Typical for 25 °C unless otherwise specified.
Specifications in *italic text* are guaranteed by design.

Power consumption

Table 1. Power consumption specifications

Not supplying power to external encoders:	
+5 V	325 mA typical, 460 mA max.
Typical supplying 1 Dynamics Research Incremental Optical Rotary Encoder part number M21AAFOBB2E 2500:	
+5 V	1058 mA typical, 1479 mA max.

Input

Table 2. Input specifications

Receiver type	SN75ALS175 quad differential receiver
Configuration	Each channel consists of PhaseA input, PhaseB input and Index input; each input switch / jumper selectable as single-ended or differential
Differential	▪ PhaseA, PhaseB and Index (+) inputs at user connector routed to (+) inputs of differential receiver. ▪ PhaseA, PhaseB and Index (-) inputs at user connector routed to (-) inputs of differential receiver.
Single - ended	▪ PhaseA, PhaseB and Index (+) inputs at user connector routed to (+) inputs of differential receiver. ▪ PhaseA, PhaseB and Index (-) inputs at user connector routed to ground. (-) inputs of differential receiver routed to +3 V reference.
Number of channels	4
Common mode input voltage range	±12 V max.
Differential input voltage range	±12 V max.
Input sensitivity	±200 mV
Input hysteresis	50 mV typ.
Input impedance	12 kΩ min.
Propagation delay	27 ns max. (tpLH, tpHL)
Absolute maximum input voltage:	
Differential	±14 V max.
Miscellaneous	▪ Meets or exceeds ANSI EIA/TIA-422-B, EIA/TIA-423-B, RS-485. ▪ Meets ITU recommendations V.10, V.11, X.26, X.27. ▪ Designed for multipoint busses on long lines and in noisy environments.

Counter

Table 3. Counter specifications

Counter type	LS7266R1 24-bit Dual-axis Quadrature Counter
Quadrature mode:	
Clock frequency	1.2 MHz max.
Separation	100 ns min.
Clock pulse width	400 ns min.
Index pulse width	300ns min.
Count mode:	
Clock frequency	30 MHz max. (25 MHz max Mod-N mode)
Clock A - high pulse width	14 ns min.
Clock A - low pulse width	14 ns min.
Filter clock (FCK)	10 MHz
Digital filter rate	10 MHz, software selectable divider (1 to 256 in single steps)
Crystal oscillator (FCK source):	
Frequency	10 MHz
Frequency accuracy	100 ppm

Interrupt controller

Table 4. Interrupt controller specifications

Controller type	8259 Programmable Interrupt Controller
Configuration	Polled mode only
Interrupts	2, 3, 5, 7, 10, 11, 12 and 15
Interrupt enable	Programmable
Interrupt sources	All Carry/Borrow outputs from LS7266R1, all Index inputs

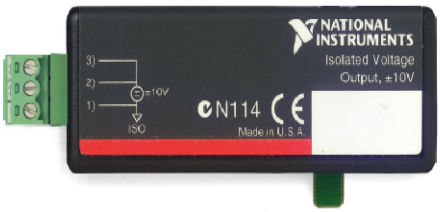
Environmental

Table 5. Environmental specifications

Operating temperature range	0 to 70 °C
Storage temperature range	-40 to 100 °C
Humidity	0 to 90% non-condensing

APPENDIX B
SIGNAL CONDITIONING ANALOG OUTPUT

NI SCC-AO10 Isolated Analog Output



Model	Channel	Description	Part Number
SCC-AO10	1	±10 V isolated voltage output	777459-50

The National Instruments SCC-AO10 is a single-channel isolated analog voltage output module with an output range of ±10 V. Each channel of the NI SCC-AO10 is referenced to its own isolated ground, allowing up to 60 VDC of common-mode voltage between grounds (channel-to-channel or channel-to-chassis). In addition, the SCC-AO10 increases the output current drive capacity of an M Series, E Series, or B Series multifunction DAQ device to ±30 mA. Because the modules are isolated, you can cascade two SCC-AO10 modules for an output range of ±20 V. There is a maximum of two SCC-AO10 modules per carrier.

Note: Use the SCC-AO10 with only multifunction DAQ devices that have analog outputs.

Specifications

SCC-AO10

Output Characteristics

Number of output channels	1 nonreferenced single ended
Input range	±10 V
Output range	±10 V
Current drive	±30 mA
Gain nonlinearity	0.5% of full-scale output range
Propagation delay	10 µs
Output noise	2.5 mV_{rms} typ; 4 mV_{rms} max
Bandwidth	>23 kHz
Slew rate	1 V/µs

Safety Isolation

Channel-to-earth (signal + common mode)	60 VDC, CAT I

Environment

Operating temperature	0 to 50 °C
Relative humidity	10 to 90% noncondensing
Stability	
Output offset temperature coefficient	300 µV/°C
Gain temperature coefficient	300 ppm/°C

Power Requirements

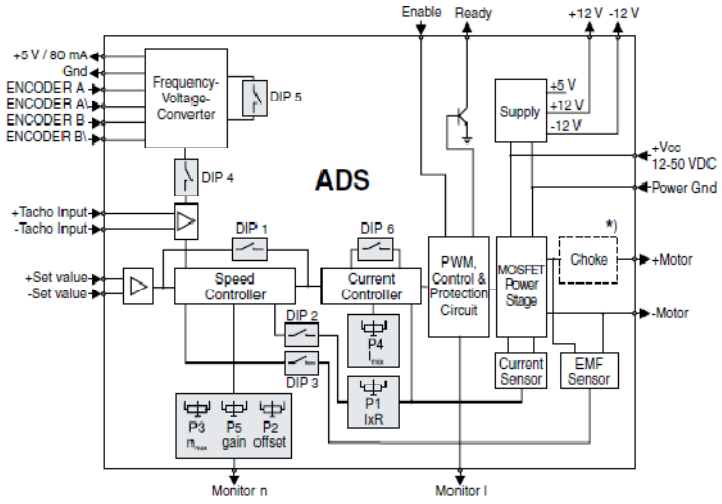
Analog	180 mW
Digital	1.15 W

APPENDIX C
MOTOR DRIVER
4-Q-DC SERVOAMPLIFIER

4-Q-DC Servoamplifier ADS

The ADS are powerful pulse width modulated (PWM) servoamplifiers for speed controlled operation of permanent magnet DC motors.
The ADS type series is distinguished by two designs and two performance classes.
The module housing is self-contained and very flexible in assembly.
The Eurocard format permits standardised rack application. The ADS performance classes cover the range between 10 and 500 Watts output power.
The ADS permits all usual operating modes: DC tacho and digital encoder speed control, IxR compensation as well as current control.
The required operating mode can be selected by DIP switch.
The ADS is protected against excess current, excess temperature and short circuit on the motor winding.
Efficiency of up to 95 % is achieved thanks to MOSFET power transistors incorporated in the servoamplifier.
Thanks to the wide input power supply range of 12 - 50 VDC, the ADS is very versatile and can be used with various power supplies.

Advantages	Features
• Comprehensive control program	ADS as STANDARD - version with 10 - 250 Watts and POWER - version with 80 - 500 Watts output power
• Suitable design	- robust metal housing in module form offers several mounting options or - standardized Eurocard format version (with accessories) for the installation in a 19"-Rack or in a plug-in card system
• Various operating modes	DC tacho speed control, digital encoder speed control, IxR compensation or current control adjustable by DIP switch
• 4-Q operation	Controlled operation during acceleration and braking in both directions
• Excellent control characteristics	Stable speed behaviour at set-value and load variations - fast current controller
• High efficiency	95 % efficiency thanks to modern MOSFET technology
• Protection circuit	Against excess current, excess temperature and short circuit of the motor leads
• Excellent price / performance ratio	Modern servoamplifier with many technical features, suitable for motors up to 500 Watts
• Outlet for reference voltage	Set value by an external potentiometer or with using external reference control voltage
• Single supply voltage	The power supply for the ADS and the encoder are made internally



*) **ADS POWER:** without built-in motor choke
ADS STANDARD: with built-in motor choke

Information on motor choke:

ADS STANDARD:
The ADS STANDARD's built in motor choke allows operation with almost all maxon DC motors with output power higher than 10 Watts without the motor heating significantly through PWM operation. The motor's continuous current must be reduced slightly if necessary.
For a few exeptions (A-max ∅26, 11 Watt as well as RE ∅25 and RE ∅26 with motor inductance lower than 350 µH) an extra inductance (choke) of at least 200 µH is necessary.

ADS POWER:
An additional external choke must be connected serially to the motor supply lead if the motor terminal inductance is less than 200 µH.
Driving a motor by ADS POWER with too low motor inductance can result in the motor overheating.
The following maxon DC motors from the stock program (from 80 Watt) need an additional external choke:

– 2260.881-51.216-200	F-Motor
– 118821	RE 75, ∅75
– 148866	RE 40, ∅40
– 148867	RE 40, ∅40

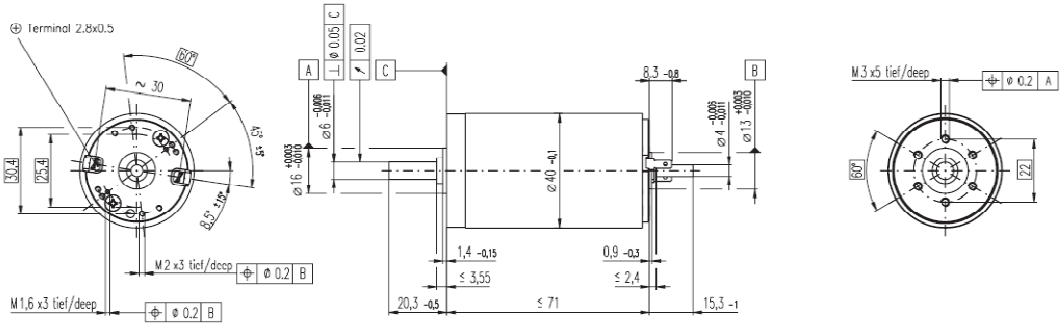
As additional terminal inductance we recommend the motor chokes available from the accessories:
– Motor choke in Eurocard format 100 x 160 [mm] (3HE, 8TE) 133350
related backplane see page 246
– Choke module 3 x 250 µH, 5 A each 90 x 70 x 49 [mm] 137303
– Choke module 3 x 150 µH, 10 A each 90 x 70 x 49 [mm] 232359

Information on encoder:

It is strongly recommended that an encoder be used with a built-in line driver for encoder operation.

APPENDIX D
MAXON DC MOTOR
RE40-40mm GRAPHITE BRUSHES 150W

RE 40 ∅40 mm, Graphite Brushes, 150 Watt



M 1:2

Stock program
Standard program
Special program (on request)

Order Number	148866	148867	148877	218008	218009	218010	218011	218012	218013	218014	218015
Motor Data											
Values at nominal voltage											
1 Nominal voltage	V	12.0	24.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0
2 No load speed	rpm	6920	7580	7580	6420	5560	3330	2690	2130	1710	1420	987
3 No load current	mA	241	137	68.6	53.7	43.7	21.9	16.7	12.5	9.67	7.77	5.16
4 Nominal speed	rpm	6370	6930	7000	5810	4920	2700	2050	1500	1080	774	339
5 Nominal torque (max. continuous torque)	mNm	94.9	170	184	183	177	187	187	189	189	188	188
6 Nominal current (max. continuous current)	A	6.00	5.77	3.12	2.62	2.20	1.38	1.12	0.898	0.721	0.593	0.413
7 Stall torque	mNm	1680	2280	2500	1990	1580	995	796	641	512	415	289
8 Starting current	A	102	75.7	41.4	28.0	19.2	7.26	4.68	3.00	1.92	1.29	0.627
9 Max. efficiency	%	88	91	92	91	91	89	88	87	86	85	83
Characteristics											
10 Terminal resistance	Ω	0.117	0.317	1.16	1.72	2.50	6.61	10.2	16.0	24.9	37.1	76.6
11 Terminal inductance	mH	0.0245	0.0823	0.329	0.460	0.612	1.70	2.62	4.14	6.40	9.31	19.2
12 Torque constant	mNm / A	16.4	30.2	60.3	71.3	82.2	137	170	214	266	321	461
13 Speed constant	rpm / V	581	317	158	134	116	69.7	56.2	44.7	35.9	29.8	20.7
14 Speed / torque gradient	rpm / mNm	4.15	3.33	3.04	3.23	3.53	3.36	3.39	3.35	3.37	3.44	3.45
15 Mechanical time constant	ms	6.03	4.81	4.39	4.36	4.35	4.31	4.31	4.31	4.31	4.32	4.33
16 Rotor inertia	gcm²	139	138	138	129	118	123	121	123	122	120	120

Specifications

Thermal data

17	Thermal resistance housing-ambient	4.65 K / W
18	Thermal resistance winding-housing	1.93 K / W
19	Thermal time constant winding	41.6 s
20	Thermal time constant motor	1120 s
21	Ambient temperature	-30 ... +100°C
22	Max. permissible winding temperature	+155°C

Mechanical data (ball bearings)

23	Max. permissible speed	12000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static) (static, shaft supported)	110 N 1200 N
28	Max. radial loading, 5 mm from flange	28 N

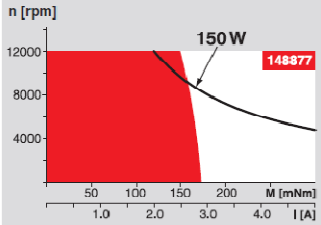
Other specifications

29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	480 g

Values listed in the table are nominal.
Explanation of the figures on page 49.

Option
Preloaded ball bearings

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

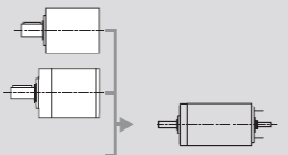
Assigned power rating

maxon Modular System

Overview on page 16 - 21

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 244

Planetary Gearhead
∅52 mm
4 - 30 Nm
Page 247



Recommended Electronics:

ADS 50/5	Page 276
ADS 50/10	277
ADS_E 50/5	277
ADS_E 50/10	277
EPOS 24/5	294
EPOS2 50/5	295
EPOS 70/10	295
EPOS P 24/5	297
Notes	18

Encoder MR
256 - 1024 CPT,
3 channels
Page 259

Encoder HED_ 5540
500 CPT,
3 channels
Page 262 / 264

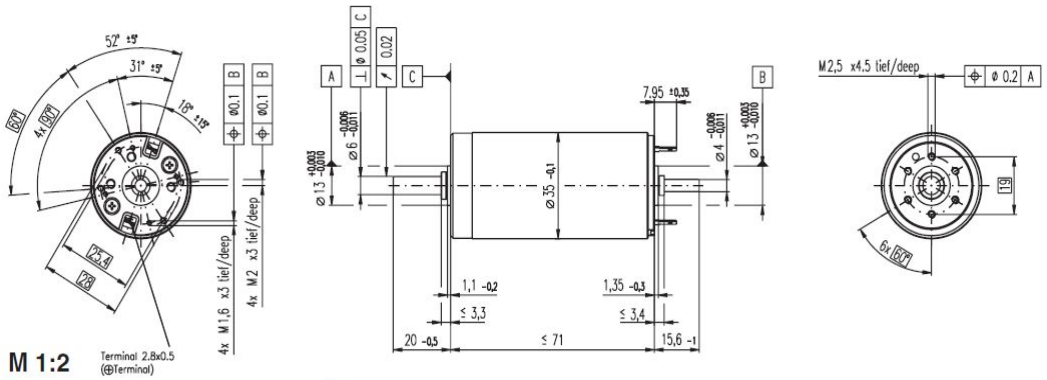
Brake AB 28
∅45 mm
24 VDC, 0.4 Nm
Page 308

Industrial Version Encoder HEDL 9140
Page 267

Brake AB 28
Page 309

APPENDIX E
MAXON DC MOTOR
RE35-35mm GRAPHITE BRUSHES 90W

RE 35 ∅35 mm, Graphite Brushes, 90 Watt



Stock program
Standard program
Special program (on request)

			Order Number												
	according to dimensional drawing		273752	323890	273753	273754	273755	273756	273757	273758	273759	273760	273761	273762	273763
	shaft length 15.7 shortened to 4 mm		285785	323891	285786	285787	285788	285789	285790	285791	285792	285793	285794	285795	285796
Motor Data															
	Values at nominal voltage														
1	Nominal voltage	V	15.0	24.0	30.0	42.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0
2	No load speed	rpm	7070	7670	7220	7530	7270	6650	5960	4740	3810	3140	2570	2100	1620
3	No load current	mA	245	168	123	92.7	77.3	68.7	59.7	44.7	34.2	27.1	21.6	17.2	12.9
4	Nominal speed	rpm	6270	6910	6420	6770	6490	5860	5150	3920	2970	2280	1710	1220	732
5	Nominal torque (max. continuous torque)	mNm	73.2	93.3	92.4	97.7	96.5	98.2	98.8	102	105	105	105	104	104
6	Nominal current (max. continuous current)	A	4.00	3.36	2.50	1.95	1.63	1.51	1.36	1.12	0.915	0.752	0.621	0.503	0.391
7	Stall torque	mNm	874	1160	949	1070	967	878	766	613	493	394	320	253	194
8	Starting current	A	45.0	39.7	24.4	20.3	15.5	12.9	10.1	6.43	4.16	2.74	1.83	1.18	0.704
9	Max. efficiency	%	81	84	84	86	85	85	84	83	82	80	79	77	74
	Characteristics														
10	Terminal resistance	Ω	0.334	0.605	1.23	2.07	3.09	3.72	4.75	7.46	11.5	17.5	26.2	40.5	68.2
11	Terminal inductance	mH	0.085	0.191	0.340	0.620	0.870	1.04	1.29	2.04	3.16	4.65	6.89	10.3	17.1
12	Torque constant	mNm / A	19.4	29.2	38.9	52.5	62.2	68	75.8	95.2	119	144	175	214	276
13	Speed constant	rpm / V	491	328	246	182	154	140	126	100	80.5	66.4	54.6	44.7	34.6
14	Speed / torque gradient	rpm / mNm	8.43	6.79	7.76	7.16	7.62	7.67	7.89	7.85	7.84	8.08	8.19	8.46	8.55
15	Mechanical time constant	ms	5.97	5.60	5.50	5.40	5.38	5.38	5.39	5.38	5.37	5.38	5.39	5.39	5.41
16	Rotor inertia	gcm²	67.6	78.7	67.6	72.0	67.4	67.0	65.2	65.4	65.5	63.6	62.8	60.8	60.4

Specifications

Thermal data

17	Thermal resistance housing-ambient	6.2 K / W
18	Thermal resistance winding-housing	2.0 K / W
19	Thermal time constant winding	30 s
20	Thermal time constant motor	1050 s
21	Ambient temperature	-30 ... +100°C
22	Max. permissible winding temperature	+155°C

Mechanical data (ball bearings)

23	Max. permissible speed	12000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static)	110 N
	(static, shaft supported)	1200 N
28	Max. radial loading, 5 mm from flange	28 N

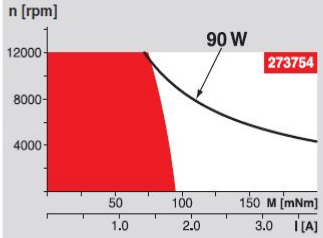
Other specifications

29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	340 g

Values listed in the table are nominal.
Explanation of the figures on page 49.

Option
Hollow shaft as special design
Preloaded ball bearings

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System

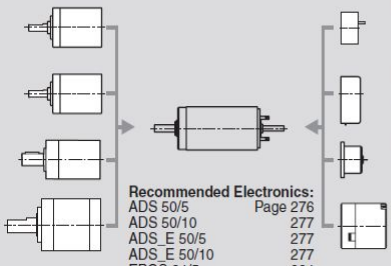
Overview on page 16 - 21

Planetary Gearhead
∅32 mm
0.75 - 4.5 Nm
Page 239

Planetary Gearhead
∅32 mm
1.0 - 6.0 Nm
Page 240

Planetary Gearhead
∅32 mm
8 Nm
Page 242

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 244



Recommended Electronics:

ADS 50/5	Page 276
ADS 50/10	277
ADS_E 50/5	277
ADS_E 50/10	277
EPOS 24/5	294
EPOS2 50/5	295
EPOS P 24/5	297
Notes	18

Encoder MR
256 - 1024 CPT,
3 channels
Page 259

Encoder HED_ 5540
500 CPT,
3 channels
Page 262 / 264

DC-Tacho DCT
∅22 mm
0.52 V
Page 271

Brake AB 28
∅40 mm
24 VDC, 0.4 Nm
Page 308

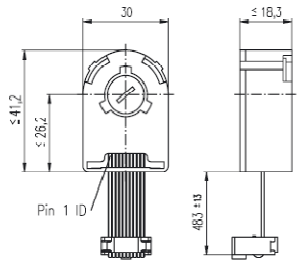
APPENDIX F
POWER SUPPLY

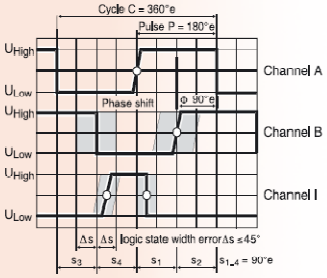
Output	
	60V, 18A
REGULATION (CV)	
Load	≤ 0.1% + 5mV
Line	≤ 0.05% + 5mV
REGULATION (CC)	
Load	≤ 0.2% + 15mA
Line	≤ 0.2% + 15mA
RIPPLE & NOISE	
Voltage (mVrms)	≤ 10mVrms
Voltage (mVp-p)	≤ 100mVp-p 20Hz~20MHz
Current (mArms)	≤ 0.2% + 40mA
RESOLUTION	
Voltage > 36V	20mV
Current 60V	10mA
PROGRAM ACCURACY	
Voltage > 36V	≤ 0.05% + 50mV
Current 60V	≤ 0.2% + 30mA
READBACK (METER) RESOLUTION	
Voltage	See Resolution
Current	See Resolution
READBACK (METER) ACCURACY	
Voltage	See Program Accuracy
Current	See Program Accuracy
READBACK TEMP. COEFFICIENT	
Voltage (25 ± 5°C)	≤ 100ppm/°C
RESPONSE (RISE/FALL) TIME	
Voltage Up (10%~90%)	≤ 150mS (≤ 95% rating load)
Voltage Down (90%~10%)	≤ 150mS (≥ 10% rating load)
RECOVERY TIME (50% Step Load Change From 20%~75%)	
CV Mode	≤ 2mS
PROTECTION	
OVP/OCP/OHP	Yes
Push Current	Yes
Output ON/OFF CONTROL	
	Yes
INTERFACE	
	Stanard : RS-232C ; Optional : GPIB (IEEE-488.2)(GPIB and RS-232C are exclusive)
POWER SOURCE	

	AC 90V ~ AC 250V, 50/60Hz
ACCESSORY	
	Power cord x 1 User manual x 1
DIMENSIONS & WEIGHT	
	268(W) x 141(H) x 388(D)mm Approx. 9.3kg

APPENDIX G
Encoder HEDL 5540

Encoder HEDL 5540, 500 - 1000 CPT, 3 Channels, with Line Driver RS 422





Stock program
Standard program
Special program (on request!)

Order Number	110512	110514	110516
Type			
Counts per turn	500	500	500
Number of channels	3	3	3
Max. operating frequency (kHz)	100	100	100
Shaft diameter (mm)	3	4	6

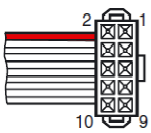
Combination

+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see: + Gearhead	
EC-max 30, 40 W	173					62.6	
EC-max 30, 40 W	173	GP 32, 1 - 6 Nm	221			●	
EC-max 30, 40 W	173			AB 20	278	94.7	
EC-max 30, 40 W	173	GP 32, 1 - 6 Nm	221	AB 20	278	●	
EC-max 30, 60 W	174					84.6	
EC-max 30, 60 W	174	GP 42, 3 - 15 Nm	225			●	
EC-max 30, 60 W	174			AB 20	278	116.7	
EC-max 30, 60 W	174	GP 42, 3 - 15 Nm	225	AB 20	278	●	
EC-power 30	175					67.6	
EC-power 30	175	GP 42, 3 - 15 Nm	225			●	
EC-power 30	175			AB 20	278	79.1	
EC-power 30	175	GP 42, 3 - 15 Nm	225	AB 20	278	●	
EC-power 30	176					84.6	
EC-power 30	176	GP 42, 3 - 15 Nm	225			●	
EC-power 30	176			AB 20	278	96.1	
EC-power 30	176	GP 42, 3 - 15 Nm	225	AB 20	278	●	
EC-max 40, 70 W	177						81.4
EC-max 40, 70 W	177	GP 42, 3 - 15 Nm	225				●
EC-max 40, 70 W	177			AB 28	281		121.4
EC-max 40, 70 W	177	GP 42, 3 - 15 Nm	225	AB 28	281		●
EC-max 40, 120 W	178						111.4
EC-max 40, 120 W	178	GP 52, 4 - 30 Nm	228				●
EC-max 40, 120 W	178			AB 28	281		151.4
EC-max 40, 120 W	178	GP 52, 4 - 30 Nm	228	AB 28	281		●

Technical Data

Supply voltage	5 V ± 10 %
Output signal	EIA Standard RS 422
drivers used:	DS26LS31
Phase shift Φ (nominal)	90°e
Logic state width s	min. 45°e
Signal rise time (typical at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	180 ns
Signal fall time (typical at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	40 ns
Index pulse width (nominal)	90°e
Operating temperature range	0 ... +70°C
Moment of inertia of code wheel	≈ 0.6 gcm²
Max. angular acceleration	250 000 rad s^{-2}
Output current per channel	min. -20 mA, max. 20 mA
Option	1000 counts per turn, 2 channel

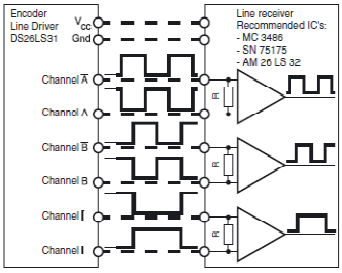
Pin Allocation



1 N.C.
2 V_{CC}
3 GND
4 N.C.
5 Channel $\overline{A}$
6 Channel A
7 Channel $\overline{B}$
8 Channel B
9 Channel $\overline{I}$ (Index)
10 Channel I (Index)

Pin type Berg 24677O
flat band cable AWG 28

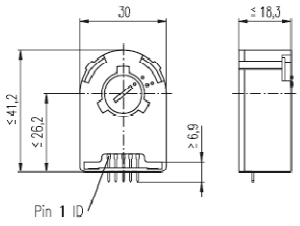
Connection example

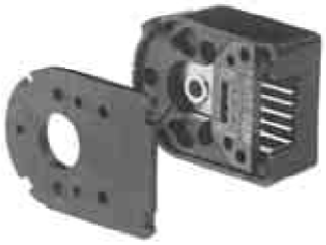


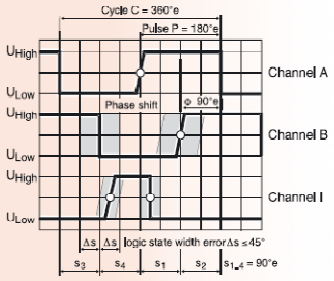
Terminal resistance R = typical 100 Ω

APPENDIX H
Encoder HEDS 5540

Encoder HEDS 5540, 500 Counts per turn, 3 Channels







Stock program
Standard program
Special program (on request!)

Order Number	110511	110513	110515
Type			
Counts per turn	500	500	500
Number of channels	3	3	3
Max. operating frequency (kHz)	100	100	100
Shaft diameter (mm)	3	4	6

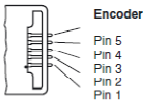
Combination

+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see: + Gearhead		
F 2260, 40 W	95						111.9	
F 2260, 40 W	95	GP 62, 8.0 - 50 Nm	229				●	
F 2260, 80 W	96						147.4	
F 2260, 80 W	96	GP 62, 8.0 - 50 Nm	229				●	
A-max 26	114-120					63.5		
A-max 26	114-120	GP 26, 0.5 - 2.0 Nm	216			●		
A-max 26	114-120	GS 30, 0.07 - 0.2 Nm	217			●		
A-max 26	114-120	GP 32, 0.75 - 6.0 Nm	218/221			●		
A-max 26	114-120	GP 32, 0.4 - 2.0 Nm	222			●		
A-max 26	114-120	GS 38, 0.1 - 0.6 Nm	223			●		
A-max 32	122/124						82.3	
A-max 32	122/124	GP 32, 0.75 - 6.0 Nm	219/221				●	
A-max 32	122/124	GS 38, 0.1 - 0.6 Nm	223				●	
EC 32, 80 W	159						76.4	
EC 32, 80 W	159	GP 32, 0.75 - 6.0 Nm	219/221				●	
EC 40, 120 W	160							88.4
EC 40, 120 W	160	GP 42, 3.0 - 15 Nm	224					●
EC 40, 120 W	160	GP 52, 4.0 - 30 Nm	227					●

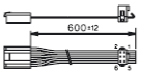
Technical Data

Supply voltage	5 V ± 10 %
Output signal	TTL compatible
Phase shift Φ (nominal)	90°e
Logic state width s	min. 45°e
Signal rise time (typical at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	180 ns
Signal fall time (typical at C_L = 25 pF, R_L = 2.7 kΩ, 25°C)	40 ns
Index pulse width (nominal)	90°e
Operating temperature range	-40 ... +100°C
Moment of inertia of code wheel	≤ 0.6 gcm²
Max. angular acceleration	250 000 rad s⁻²
Output current per channel	min. -1 mA, max. 5 mA

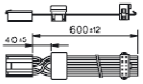
Pin Allocation



Encoder	Designation	Pin no. from 3409.506
Pin 5	Channel B	1
Pin 4	V_{CC}	2
Pin 3	Channel A	3
Pin 2	Channel I	4
Pin 1	GND	5

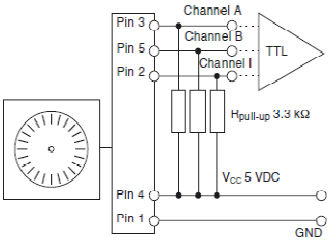


Cable with plug:
maxon Art. No. 3409.506
The plug (Harting 918.906.6803) can be fixed in the required position.



Cable with plug: (compatible with Encoder HEDS5010)
maxon Art. No. 3409.504
The plug (3M 891100101) can be fixed in the required position.

Connection example



Ambient temperature range ϑ_U – 25°C

APPENDIX I

GANTT CHART FOR SEMESTER 1

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Topic Search															
Topic Study & Problem Statement															
Proposal															
Prototype Design															
Prototype Fabrication															
Simulink Test															
Presentation															
Report															

APPENDIX J

GANTT CHART FOR SEMESTER 2

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Interface Software and Hardware															
Design Controller															
Experimental (PID Tuning)															
Result Analysis															
Thesis															
Presentation															
Thesis Complition															